PayPal™

Express Checkout Integration Guide

Last updated: October 2009

Express Checkout Integration Guide

Document Number: 100010.en_US-200910

© 2009 PayPal, Inc. All rights reserved. PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners.
The information in this document belongs to PayPal, Inc. It may not be used, reproduced or disclosed without the written approval of PayPal, Inc.
Copyright © PayPal. All rights reserved. PayPal (Europe) S.à r.l. et Cie., S.C.A., Société en Commandite par Actions. Registered office: 22-24 Boulevard Royal, L-2449, Luxembourg, R.C.S. Luxembourg B 118 349.
Consumer advisory: The PayPal™ payment service is regarded as a stored value facility under Singapore law. As such, it does not require the approval of the Monetary Authority of Singapore. You are advised to read the terms and conditions carefully.

Notice of non-liability:
PayPal, Inc. is providing the information in this document to you “AS-IS” with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.

Contents

Preface 7

Intended Audience 7
Where to Go for More Information 7
Revision History 7

Chapter 1 Introducing Express Checkout 9

The Express Checkout Experience 9
Getting Started 10
Configuring and Customizing the Express Checkout Experience 11
Additional PayPal API Operations 11
Express Checkout Flow 12
Checkout Entry Point 12
Payment Option Entry Point 12
Complete Express Checkout Flow 13
Express Checkout Building Blocks 14
Express Checkout Buttons 14
Express Checkout API Operations 15
Express Checkout Command 15
Express Checkout Token Usage 16

Chapter 2 Express Checkout Button and Logo Image Integration . . . 17

About PayPal Button and Logo Images 17
Express Checkout Images 17
Express Checkout Image Flavors 17
Dynamic Images 18
Configuring the Dynamic Image 18
Set Up the Default Image 19
Set Up Image for Dynamic Use 19
Change the Locale 20
Feedback to Buyer Meeting an Incentive 20
Choose the Image 20

Dynamic Image Command Reference 20
Dynamic Image Parameters 21
Locale Codes 21
Static PayPal Buttons and Images Source Requirements 23

Chapter 3 PayPal Name-Value Pair API Basics 25

PayPal API Client-Server Architecture 25
PayPal Name-Value Pair API Requests and Responses 25
Multiple API Operations 26
Obtaining API Credentials 28
Creating an API Signature 28
Creating an API Certificate 30
Creating an NVP Request 30
Specifying the PayPal API Operation 30
Specifying an API Credential 31
URL Encoding 32
List Syntax for Name-Value Pairs 32
Executing NVP API Operations 33
Specifying a PayPal Server 33
Logging API Operations 33
Responding to an NVP Response 34
Common Response Fields. 34
URL Decoding 34

Chapter 4 Implementing the Simplest Express Checkout Integration . 35

Setting Up the Express Checkout Transaction. 35
Obtaining Express Checkout Transaction Details 37
Completing the Express Checkout Transaction 37

Chapter 5 Testing an Express Checkout Integration 39

Chapter 6 Customizing Express Checkout 45

PayPal Review Page Order Details 45
Special Instructions to Merchant 47
Integrating Order Details into the Express Checkout Flow 48
eBay-Issued Incentives 51
Integrating eBay Incentives into the Express Checkout Flow 52

PayPal Page Style 54
Custom Page Style 55
Individual Page Style Characteristics 55
Changing the Locale 58
Handling Shipping Addresses 59
Confirmed Address 60
Suppressing the Buyer's Shipping Address 61
Shipping Address Override 63
Automatically Filling Out the PayPal Login Page 65
User Confirms Order on PayPal 67

Chapter 7 Implementing the Instant Update API 69

About the Instant Update API 69
Integration Steps 69
Post-Integration Checkout Experience 70
How the Callback Works in the Express Checkout Flow 72
Following Instant Update API Best Practices 73
Setting Up the Callback 74
GetExpressCheckoutDetails and DoExpressCheckoutPayment Changes 75
Other Considerations 75
Using the Callback 76
SetExpressCheckout 77
Callback Request 79
Callback Response 79

Chapter 8 Immediate Payment 81

Overview of Immediate Payment 81
How Immediate Payment Works in the Express Checkout Flow 81
Integrating Immediate Payment into the Express Checkout Flow 83
The Call to SetExpressCheckout 83
The Call to DoExpressCheckoutPayment 84

Chapter 9 Handling Payment Settlements 85

Sale Payment Action 85
Authorization Payment Action 85
Order Payment Action 86

Chapter 10 Handling Recurring Payments 89

How Recurring Payments Work 89
Limitations 90
Recurring Payments Terms 90
Recurring Payments With Express Checkout 91
Initiating the Processing Flow With SetExpressCheckout 93
Redirecting the Buyer's Browser to PayPal 94
Getting Buyer Details Using GetExpressCheckoutDetails 95
Creating the Profiles With CreateRecurringPaymentsProfile 95
Options for Creating a Recurring Payments Profile 95
Specifying the Regular Payment Period 95
Including an Optional Trial Period 96
Specifying an Initial Payment 97
Maximum Number of Failed Payments 97
Billing the Outstanding Amount 98
Recurring Payments Profile Status 98
Getting Recurring Payments Profile Information 98
Modifying a Recurring Payments Profile 99
Updating Addresses 100
Updating the Billing Amount 100
Billing the Outstanding Amount of a Profile 100
Recurring Payments Notifications 101

Chapter 11 Using Other PayPal API Operations 103

Issuing Refunds 103
Handling Payment Review 104

Chapter 12 Integrating giropay with Express Checkout 107

giropay Page Flows 107
giropay Payment Page Flow 107
Cancelled or Unsuccessful giropay Payment Page Flow 108
giropay Integration 109
Initiate the Flow with SetExpressCheckout 109
Redirect the Customer to PayPal 109
Complete the Transaction 110
Receive Transaction Status Notification 110

Preface

This document describes Express Checkout integration.

Intended Audience

This document is intended for developers and merchants implementing Express Checkout.

Where to Go for More Information

For information on the administrative tasks you can perform from your PayPal account, see the *Merchant Setup and Administration Guide*. The guide is available on the Documentation page linked to the Library tab in Developer Central.

Revision History

Revision history for *Express Checkout Integration Guide*.

TABLE 1.1 Revision history

Date	Description
10/05/2009	Added Immediate Payment. Edited for technical accuracy. Removed PayPal placement guidelines.
06/30/2009	Added a section on payment review.
6/04/2009	Added a chapter on pre-populating the PayPal review page. Updated PayPal Review pages. Moved some customization topics out of this guide. They are now in the *Merchant Setup and Administration Guide*.
04/08/2009	Added a chapter describing the Instant Update Callback API.
03/03/2009	Updated to allow useraction=continue for eBay incentives.
11/13/2008	Added information about integrating dynamic images and added information about order details that can be displayed on the PayPal Review page.
06/30/2008	Complete revision.

1 Introducing Express Checkout

Express Checkout is PayPal's premier checkout solution, which streamlines the checkout process for buyers and keeps them on the merchant's site after making a purchase.

- The Express Checkout Experience
- Getting Started
- Express Checkout Flow
- Express Checkout Building Blocks

NOTE: For information about administrative tasks you can perform from your PayPal account such as adding users, setting up custom page styles, and managing multiple currency balances, see the Merchant Setup and Administration Guide. The guide is located on the Documentation page linked to the Library tab on Developer Central.

The Express Checkout Experience

Express Checkout makes it easier for your customers to pay and allows you to accept PayPal while retaining control of the buyer and overall checkout flow.

Consider your buyer's experience before implementing Express Checkout. A generic flow probably has the following sequence of pages:

A generic checkout flow

In a typical checkout flow, the buyer

1. Checks out from the shopping cart page
2. Provides shipping information
3. Chooses a payment option and provides billing and payment information
4. Reviews the order and pays
5. Receives an order confirmation

In an Express Checkout flow, a buyer still checks out at the beginning of the flow and pays on your site; however, the buyer does not reenter shipping, billing, or payment information, because it is already available from PayPal. This simplifies and expedites the checkout

process. The buyer can then proceed to review the order on your site. You can include other necessary checkout steps. You can also up-sell to the buyer on your Review Order page.

The following diagram shows the Express Checkout flow:

Express Checkout Flow

In the Express Checkout flow, the buyer

1. Chooses Express Checkout by clicking the **Check out with PayPal** button
2. Logs into PayPal to authenticate his or her identity
3. Reviews the transaction on PayPal
4. Confirms the order and pays from your site
5. Receives an order confirmation

Getting Started

You can implement Express Checkout in just four steps.

1. Place PayPal checkout buttons and PayPal payment mark images in your checkout flow.
2. For each PayPal button that you place, modify your page to handle the button click.

   Use a PayPal API Express Checkout operation to set up the interaction with PayPal and redirect the browser to PayPal to initiate buyer approval for the payment.
3. On your order confirmation page, obtain the payment authorization from PayPal and use PayPal API Express Checkout operations to obtain the shipping address and accept the payment.
4. Test your integration using the PayPal sandbox.

Because PayPal offers you the flexibility and opportunity to control your checkout process, you should understand how your current checkout flow works and become familiar with the Express Checkout flow. Start by reviewing Express Checkout Flow. For additional background information that will help you get started, see Express Checkout Building Blocks.

Configuring and Customizing the Express Checkout Experience

After you implement and test your basic Express Checkout integration, you should evaluate the additional features provided by Express Checkout.

Express Checkout can be configured and customized to fit into your site. Carefully evaluate each option because the more you do to streamline the checkout process and make Express Checkout seamless to your customers, the more likely your sales will increase.

At a minimum, you should set your logo on the PayPal site and provide order details in the transaction history. If you do not need the benefits associated with paying on your site, consider using the PayPal confirmation page as your Order Review page to further streamline the user experience. This can lead to better order completion, also know as *conversion*, rate.

You can configure the look and feel of PayPal pages to match the look and feel of your site by specifying the

- Logo to display
- Colors for the background and border
- Language in which PayPal content is displayed

You should include

- Order details, including shipping and tax, during checkout

  IMPORTANT: Not displaying this information is a major cause of shopping cart abandonment during checkout.

- Shipping information for non-digital goods, which can be your address information for the buyer or the address on file with PayPal; if you use the address on file with PayPal, you can specify whether or not it must be a confirmed address

You can also handle special situations, including

- Associating a payment with an eBay auction item or an invoice number
- Accepting payments with giropay

Additional PayPal API Operations

You can use PayPal API operations to include advanced processing and “back office” processes with Express Checkout, such as

- Capturing payments associated with authorizations and orders
- Handling recurring payments
- Issuing refunds, providing a transaction search capability, and providing other “back office” operations

Express Checkout Flow

You initiate the Express Checkout flow on your shopping cart page and on your payment options page.

Express Checkout consists of one flow with two entry points:

- At the beginning of the checkout flow; the buyer clicks the **Checkout with PayPal** button
- Upon payment; the buyer selects PayPal as a payment option

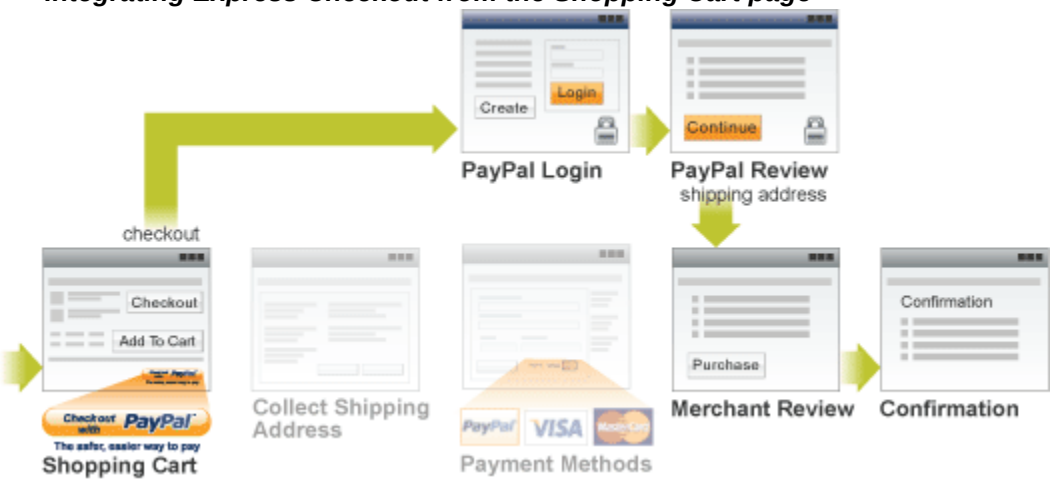
You must integrate Express Checkout from both entry points. They are explained separately for simplicity.

Checkout Entry Point

Buyers initiate the Express Checkout flow on your shopping cart page by clicking the **Checkout with PayPal** button.

The following diagram shows how Express Checkout integrates with a typical checkout flow:

Integrating Express Checkout from the Shopping Cart page



Payment Option Entry Point

Buyers initiate the Express Checkout flow on your payment methods page by selecting PayPal as the default option.

The following diagram shows how to integrate Express Checkout from your payment methods page:

Integrating Express Checkout from the Payment Methodss page

Note that if your checkout flow omits the Merchant Review page and proceeds directly to your Confirmation page, you can change the text on the PayPal Review page from Continue to Pay Now. For details on how to change the text on the PayPal Review page from `'Continue'` to `'Pay Now'` see "User Confirms Order on PayPal" on page 67.

Complete Express Checkout Flow

To implement Express Checkout, you must offer it both as a checkout option and as a payment method.

You add Express Checkout to your existing flow by placing the **Check out with PayPal** button on your checkout page and by providing the **PayPal** mark on your billing payment methods page. The following diagram shows the complete flow:

Complete Express Checkout flow

To implement the complete Express Checkout flow

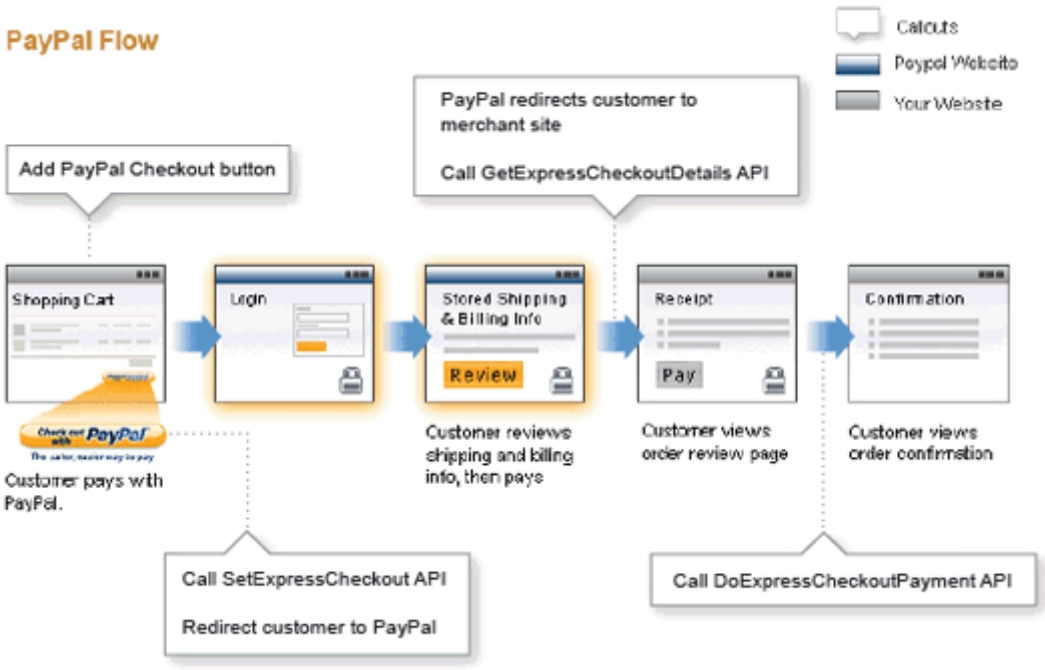
- On your Shopping Cart page, place the **Check out with PayPal** button and handle button clicks by setting up the Express Checkout transaction and redirecting your buyer's browser to PayPal.
- On your Payment Methods page, associate the **PayPal** mark with an option. Handle selection of the **PayPal** mark by setting up the Express Checkout transaction and redirecting your buyer's browser to PayPal.
- After returning from PayPal, complete the Express Checkout transaction by obtaining shipping information from PayPal and accepting the payment.

Express Checkout Building Blocks

You implement Express Checkout flows with Express Checkout buttons, PayPal API operations, PayPal commands, and tokens.

The following conceptual diagram identifies the building blocks that you use to integrate Express Checkout on your website:

Express Checkout Integration



NOTE: Tokens are not shown in the diagram.

Express Checkout Buttons

PayPal provides buttons and images for you to place on your website.

To implement Express Checkout, you must place the following button on your Shopping Cart page:

To implement PayPal as a payment option, which is part of Express Checkout, associate the PayPal mark image with the option. Using a radio button, as shown in the image below, is the recommended way to do this:

Express Checkout API Operations

The PayPal API provides three operations for Express Checkout, one to set up the transaction, one to obtain information about the buyer, and one to handle the payment and complete the transaction.

API Operation	Description
`SetExpressCheckout`	Sets up the Express Checkout transaction. You can specify information to customize the look and feel of the PayPal site and the information it displays. You must include the following information: • URL to the page on your website that PayPal redirects to after the buyer successfully logs into PayPal and approves the payment. • URL to the page on your website that PayPal redirects to if the buyer cancels. • Total amount of the order or your best estimate of the total. It should be as accurate as possible.
`GetExpressCheckout`	Obtains information about the buyer from PayPal, including shipping information.
`DoExpressCheckoutPayment`	Completes the Express Checkout transaction, including the actual total amount of the order.

Express Checkout Command

PayPal provides a command that you use when redirecting to PayPal, which enables your buyer to log into PayPal to approve an Express Checkout payment.

When you redirect your buyer's browser to PayPal, you must specify `_ExpressCheckout` command for Express Checkout. You also specify the token that identifies the transaction, which was returned by the `SetExpressCheckout` API operation.

IMPORTANT: To enable PayPal to redirect back to your website, you must have already invoked the `SetExpressCheckout` API operation, specifying URLs that PayPal uses to redirect back to your site. PayPal redirects to the *success* URL when the buyer pays on PayPal; otherwise, PayPal redirects to the *cancel* URL.

If the buyer approves the payment, PayPal redirects to the success URL with the following information:

- The token that was included in the redirect to PayPal
- The buyer's unique identifier (Payer ID)

If the buyer cancels, PayPal redirects to the cancel URL with the token that was included in the redirect to PayPal.

Express Checkout Token Usage

Express Checkout uses a token to control access to PayPal and execute Express Checkout API operations.

The `SetExpressCheckout` API operation returns a token, which is used by other Express Checkout API operations and by the `_ExpressCheckout` command to identify the transaction. The life of the token is approximately three hours.

2 Express Checkout Button and Logo Image Integration

PayPal hosts the PayPal button and logo images that you use on your website. Using PayPal's buttons and logos is convenient and standardizes appearance on websites that use PayPal as a payment option.

- About PayPal Button and Logo Images
- Dynamic Images
- Configuring the Dynamic Image
- Dynamic Image Command Reference
- Static PayPal Buttons and Images Source Requirements

About PayPal Button and Logo Images

To inform buyers that PayPal is accepted on your website, you must place PayPal button and logo images in your checkout flow.

PayPal Express Checkout requires that you integrate two images. The Check out with PayPal button and the PayPal Acceptance mark.

Express Checkout Images

The Check out with PayPal button is the image you place on your shopping cart page. The US version of the image looks like this. PayPal also provides buttons for other locales.

The PayPal Acceptance Mark is the image you place on your payment methods page. It looks like this:

Express Checkout Image Flavors

The Check out with PayPal button and the PayPal Acceptance mark images are available in two flavors:

- Dynamic image

- Static image

The dynamic images enable PayPal to change their appearance dynamically. If, for example, you have signed up to participate in a PayPal campaign, PayPal can change the appearance of the image dynamically for the duration of that campaign based on parameter information you append to the image URL. By default, the Express Checkout images appears as shown above.

The static images cannot be changed dynamically. To participate in a PayPal campaign, you would have to manually update the image code to change the image displayed and restore the default image when the campaign is over. The only way you can have image management taken care of for you is to replace static images in your implementation with dynamic images.

Dynamic Images

To use dynamic images, you must pass information to PayPal as parameters appended to the image URL. Your unique ID tells PayPal whether or not you are participating in events that require image changes. Other information you pass instructs PayPal on the types of images to return.

If, for example, you are participating in a PayPal campaign that you have signed up for with PayPal and you have passed the appropriate parameter information to PayPal, PayPal automatically updates the image to reflect the campaign information. When the campaign is over, PayPal restores the default image. You are not responsible for scheduling or making changes to your website application code before, during, or after the campaign. It is all handled for you when you set up the dynamic image.

If you require localized campaign images, you can have the localized button image display for each country in which you participate. Simply assign the correct code for the country to the locale parameter you append to the dynamic image URL. PayPal will return to the default button image associated with each locale when the campaign is not available.

Configuring the Dynamic Image

To set up the dynamic image, you provide the name-value pair parameter information in the image URL. You can pass information in the image URL for any of the following options.

- Set Up the Default Image
- Set Up Image for Dynamic Use
- Change the Locale
- Feedback to Buyer Meeting an Incentive
- Choose the Image

Set Up the Default Image

The following URL points to the default Check out with PayPal image:

https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image

To make the image dynamic, you need only add parameters to this URL to specify the changes you want displayed.

To test in the Sandbox environment, send the image to the following Sandbox URL:

https://fpdbs.sandbox.paypal.com/dynamicimageweb?cmd=_dynamic-image

Set Up Image for Dynamic Use

To set up the image URL for dynamic use, you associate it with your PayPal merchant account number or `pal`. You can obtain your pal by getting it from the Profile page, contacting PayPal, or calling the `GetPalDetails` API.

This is an example call to `GetPalDetails` request.

Request Parameters:

[requiredSecurityParameters]
`&METHOD=GetPalDetails`

Response Parameters

This `GetPalDetails` response returns the value of `PAL` and your country code (`LOCALE`), as shown below:

[successResponseFields]
`&PAL=SFJCXFDLNFR5U`
`&LOCALE=en_US`

1. Append the `pal` parameter to the image URL, and set the parameter to the value of your encrypted PayPal merchant account number.

   ```
   https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image&pal=SFJCXFDLNFR5U
   ```

2. You can optionally change the value of `LOCALE`. See Change the Locale for details.

3. Place the URL with parameter information at the appropriate image locations in your web application.

   The `pal` alerts PayPal to campaigns in which you are participating. PayPal obtains this information from your account and replaces the default image with the appropriate campaign image during that campaign.

NOTE: If you pass in a `pal` value matching a merchant account that is not yours, PayPal displays the image for that account. Be sure to pass the `pal` value matching your account.

Change the Locale

To specify the locale of the image, append the `locale` parameter set to the code for the appropriate country to the image URL. If a country does not have a localized image or if you do not pass a `locale` value, the default US image displays. This example displays the image for the Spanish locale:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-
image&pal=SFJCXFDLNFR5U&locale=es_ES
```

If you are participating in a campaign across multiple countries, you can set the image locale for each country in which you participate. PayPal returns the default image associated with the locale when the campaign is over.

Feedback to Buyer Meeting an Incentive

Pass the order total amount in the `ordertotal` parameter so PayPal can determine if the buyer is eligible for an incentive. Say, for example, that you are participating in a campaign in which the buyer is eligible for a 20% discount when their order meets a minimum of $50.00. You can pass that value to PayPal in the `ordertotal` parameter, as shown here:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-
image&pal=SFJCXFDLNFR5U&ordertotal=50.00
```

When a buyer's order meets or exceeds $50.00, PayPal displays the incentive image informing the buyer of their eligibility for the discount. When a buyer's order is less than $50.00, PayPal displays the default image.

NOTE: If `ordertotal` is not passed, PayPal does not display the incentive image even if the buyer is eligible for the incentive.

Choose the Image

To specify the image that you want to display, set the value of `buttontype`. This example sets `buttontype` to the PayPal Acceptance Mark image:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-
image&pal=SFJCXFDLNFR5U&buttontype=ecmark
```

The default value for `buttontype` is `ecshortcut`.

Dynamic Image Command Reference

To set up the information that enables dynamic images, you add name-value pairs to the dynamic image URL. Parameters and values are described below.

Dynamic Image Parameters

The table below describes the dynamic image name-value pair parameters.

Dynamic-Image Command Variable Descriptions

`pal`	Type: encrypted PayPal account number *(Optional)* Unique identification number. When merchants sign up for a PayPal business account, PayPal assigns them an account number. The `pal` value represents the pay-to merchant account, not a third party making the API request on behalf of this merchant. **NOTE:** If `pal` is not passed, PayPal displays the default Check out with PayPal button.
`ordertotal`	Type: numeric *(Optional)* The total cost of the order to the buyer. If shipping and sales tax are known, include them in this value. If not, this value should be the current subtotal of the order. **NOTE:** If `ordertotal` is not passed, PayPal does not display the incentive image even if the buyer is eligible for the incentive. Character length and limitations: Must not exceed $10,000.00 USD in any currency. No currency symbol.Must have two decimal places, decimal separator must be a period (.), and the optional thousands separator must be a comma(,).
`locale`	Type: string *(Optional)* The five-character locale code. See Locale Codes. Any other values default to US. **NOTE:** The merchant can participate in one campaign per country.
`buttontype`	Type: string *(Optional)* Indicates a dynamic image. The values are: • (Default) Check out with PayPal button image: `ecshortcut` • PayPal Acceptance Mark image: `ecmark`

Locale Codes

The table below lists the locale values. Country code is the two-letter code for the country. Language priority is the language associated with the country code where language_0 is the default.

Country codes, language priorities, and locale values

Country code	Language priority	Locale
AT	language_0	de_DE
AT	language_1	en_US
AU	language_0	en_AU
BE	language_0	en_US
BE	language_1	nl_NL
BE	language_2	fr_FR
C2	language_0	en_US
C2	language_1	zh_XC
C2	language_2	fr_XC
C2	language_3	es_XC
CH	language_0	de_DE
CH	language_1	fr_FR
CH	language_2	en_US
CN	language_0	zh_CN
default	language_0	en_US
default	language_1	fr_XC
default	language_2	es_XC
default	language_3	zh_XC
DE	language_0	de_DE
DE	language_1	en_US
ES	language_0	es_ES
ES	language_1	en_US
FR	language_0	fr_FR
FR	language_1	en_US
GB	language_0	en_GB
GF	language_0	fr_FR
GF	language_1	en_US
GI	language_0	en_US
GP	language_0	fr_FR
GP	language_1	en_US

Country code	Language priority	Locale
IE	language_0	en_US
IT	language_0	it_IT
IT	language_1	en_US
JP	language_0	ja_JP
JP	language_1	en_US
MQ	language_0	fr_FR
MQ	language_1	en_US
NL	language_0	nl_NL
NL	language_1	en_US
PL	language_0	pl_PL
PL	language_1	en_US
RE	language_0	fr_FR
RE	language_1	en_US
US	language_0	en_US
US	language_1	fr_XC
US	language_2	es_XC
US	language_3	zh_XC

Static PayPal Buttons and Images Source Requirements

Using the static image code on the PayPal servers eliminates the need for you to maintain them yourself.

PayPal requires that you use Express Checkout images hosted on PayPal's secure servers, rather than hosting copies of these images on your own servers. Out-of-date PayPal buttons on you site could reduce customer confidence. If the buttons are updated, the new buttons will appear automatically in your application.

To get the HTML code for the Check out with PayPal button, use the URLs in the table below.

Check out with PayPal button URLs

Country	Image URL
Australia	https://www.paypal.com/au/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Austria	https://www.paypal.com/at/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Belgium	https://www.paypal.com/be/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Canada	https://www.paypal.com/ca/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
China	https://www.paypal.com/cn/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
France	https://www.paypal.com/fr/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Germany	https://www.paypal.com/de/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Italy	https://www.paypal.com/it/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Japan	https://www.paypal.com/j1/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Netherlands	https://www.paypal.com/nl/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Poland	https://www.paypal.com/pl/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Spain	https://www.paypal.com/es/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
Switzerland	https://www.paypal.com/ch/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
United Kingdom	https://www.paypal.com/uk/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside
United States	https://www.paypal.com/us/cgi-bin/webscr?cmd=xpt/Merchant/merchant/ExpressCheckoutButtonCode-outside

To get the HTML code for the PayPal Acceptance Mark image and for other general information about images, change the command value in the URLs above to:
`cmd=xpt/Marketing_CommandDriven/general/OnlineLogoCenter-outside`

For example, to get the PayPal Acceptance Mark image for Poland, change the URL to:
https://www.paypal.com/pl/cgi-bin/webscr?cmd=xpt/Marketing_CommandDriven/general/OnlineLogoCenter-outside

3 PayPal Name-Value Pair API Basics

The PayPal API uses a client-server model in which your site is a client of the PayPal server.

- PayPal API Client-Server Architecture
- Obtaining API Credentials
- Creating an NVP Request
- Executing NVP API Operations
- Responding to an NVP Response

PayPal API Client-Server Architecture

The PayPal API uses a client-server model in which your site is a client of the PayPal server.

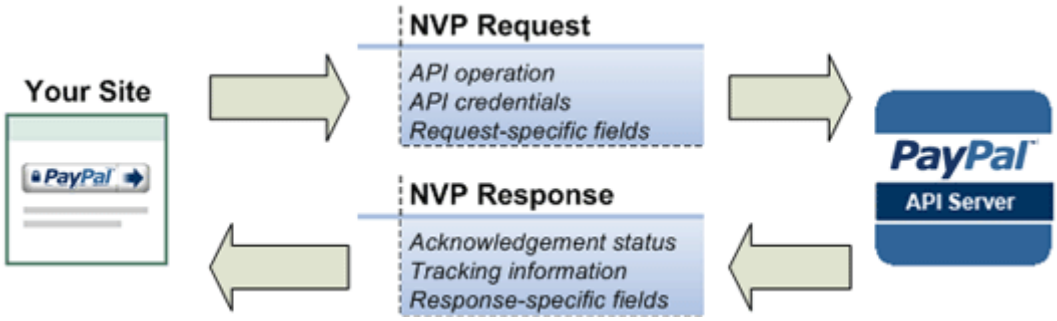
You set up web pages on your site that initiate actions on a PayPal API server by sending a request to the server. The PayPal server responds with a confirmation that the requested action was taken or that an error occurred. The response may contain additional information related to the request. The following diagram shows the basic request-response mechanism.

For example, you might want to obtain the customer's shipping address from PayPal. You could initiate a request that specifies an API operation that gets customer details. The response from the PayPal API server would contain information about whether the request was successful. If the operation was successful, the response would contain the requested information; in this case, the customer's shipping address. If the operation fails, there will be one or more error messages.

PayPal Name-Value Pair API Requests and Responses

To perform a PayPal NVP API operation, you send an NVP-formatted request to a PayPal NVP server and interpret the response.

In the following diagram, a request is generated on your site. At the PayPal server, the request is executed, and the response is returned to your site.



The request identifies

- The name of the API operation to be performed and its version; for example, SetExpressCheckout for version 52.0
- Credentials that identify the PayPal account making the request
- Request-specific information that controls the API operation to be performed

A PayPal API server performs the operation and returns a response. The response contains

- An acknowledgement status that indicates whether the operation was a success or failure and whether any warning messages were returned
- Information that can be used by PayPal to track execution of the API operation
- Response-specific information required to fulfill the request

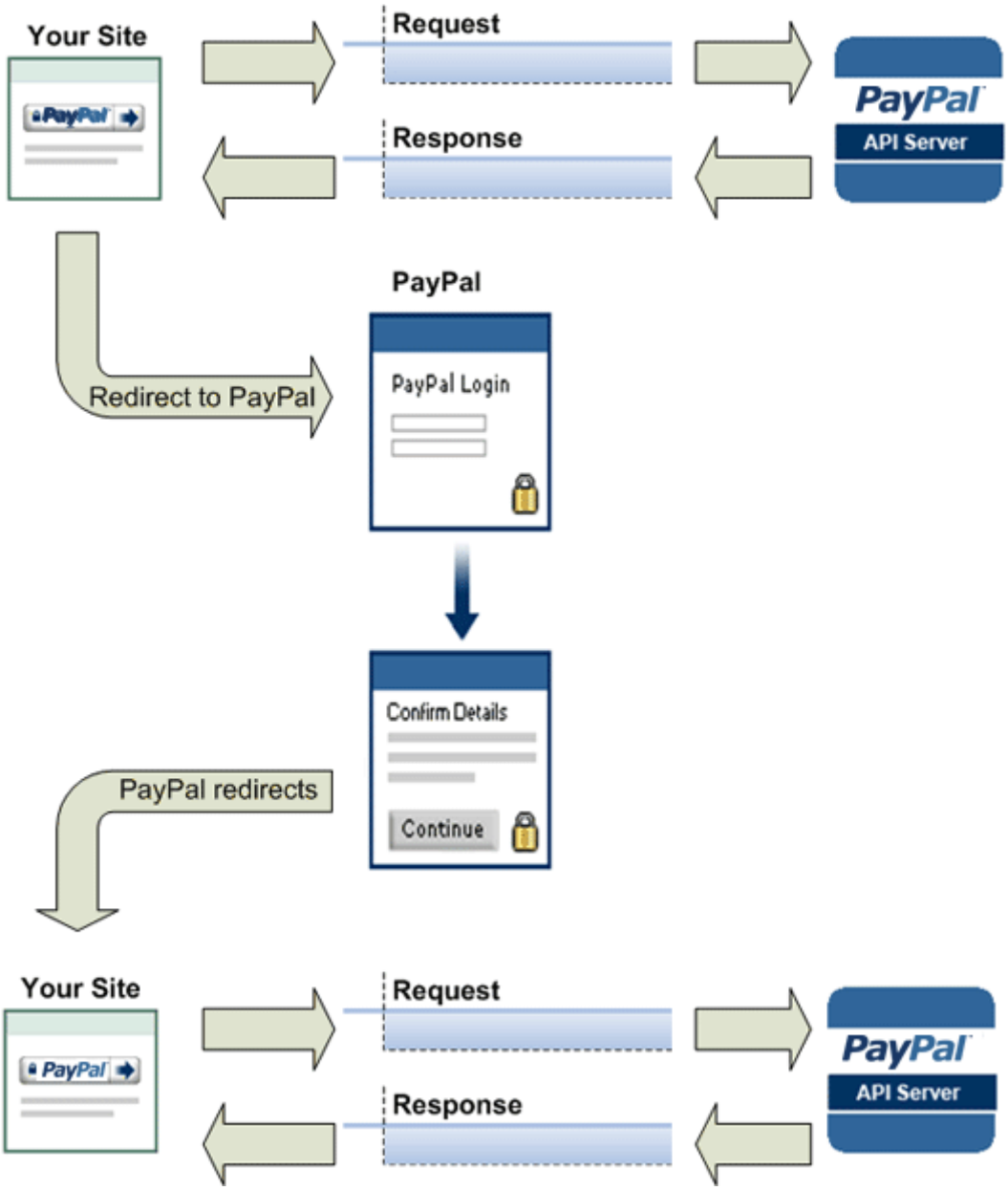
Multiple API Operations

Some of the features you want to implement require multiple API operations.

Features, such as Express Checkout, require more than one API operation. Typically, these features require you to

1. Invoke an API operation, such as `SetExpressCheckout`, that sets up the return URL to which PayPal redirects your buyer's browser after the buyer finishes on PayPal. Other setup also may be performed.
2. Invoke additional API operations after receiving the buyer's permission on PayPal, for example, `GetExpressCheckoutDetails` or `DoExpressCheckoutPayment`.

The following diagram shows the execution flow between your site and PayPal:



Token Security

Typically, the API operation that sets up a redirection to PayPal returns a token, which is passed as a parameter in the redirect to PayPal. The token also may be required in related API operations.

Obtaining API Credentials

To use the PayPal API, you must have API credentials that identify you as a PayPal business account holder authorized to perform various API operations.

Although you can use either an API signature or a certificate for credentials, PayPal recommends you use a signature.

IMPORTANT: You can use either a signature or a certificate; however, you cannot use a signature and a certificate at the same time.

Creating an API Signature

You must establish credentials to use the PayPal API; typically, you create an API signature.

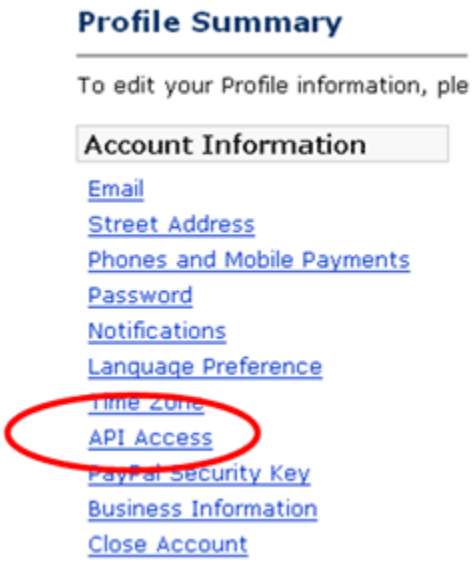
You must have a PayPal business account.

An API signature is a credential that consists of an API username along with an associated API password and signature, all of which are assigned by PayPal.

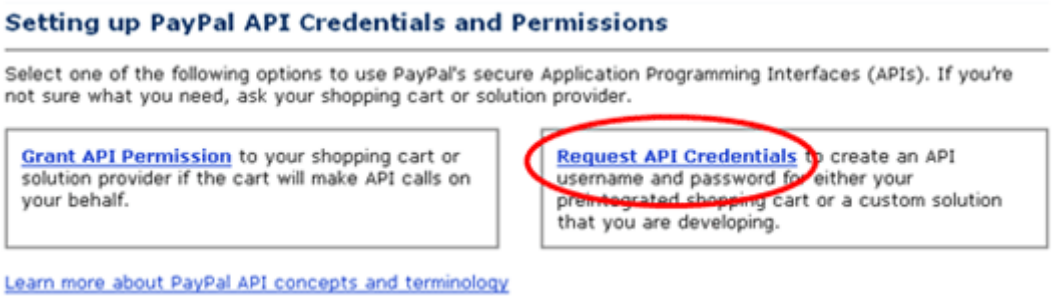
To create an API signature

1. Log into PayPal and click **Profile**.

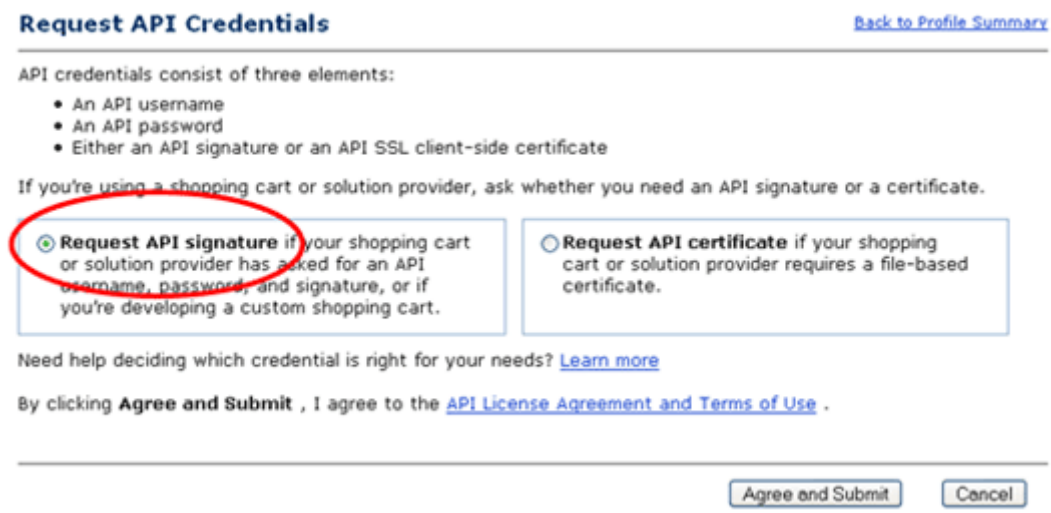
2. Click **API Access** from the Profile menu.



3. Click **Request API Credentials**.


Setting up PayPal API Credentials and Permissions
Select one of the following options to use PayPal's secure Application Programming Interfaces (APIs). If you're not sure what you need, ask your shopping cart or solution provider.
Grant API Permission to your shopping cart or solution provider if the cart will make API calls on your behalf.
Request API Credentials to create an API username and password for either your pre-integrated shopping cart or a custom solution that you are developing.
Learn more about PayPal API concepts and terminology

4. Check **Request API signature** and click **Agree and Submit**.


Request API Credentials
Back to Profile Summary
API credentials consist of three elements:
An API username
An API password
Either an API signature or an API SSL client-side certificate
If you're using a shopping cart or solution provider, ask whether you need an API signature or a certificate.
Request API signature if your shopping cart or solution provider has asked for an API username, password, and signature, or if you're developing a custom shopping cart.
Request API certificate if your shopping cart or solution provider requires a file-based certificate.
Need help deciding which credential is right for your needs? Learn more
By clicking Agree and Submit , I agree to the API License Agreement and Terms of Use .
Agree and Submit
Cancel

5. To complete the process, click **Done**.

View or Remove API Signature
Back to Profile Summary
For preconfigured shopping carts: Copy and paste the API username, password, and signature into your shopping cart configuration or administration screen.
For building custom shopping carts: Store the following credential information in a secure location with limited access.
Credential: API Signature
API Username: smith011_api1.verizon.net
API Password: VL4SAZXXDD7R8S6P
Signature: AkaqkMYp-zjdBgDMXEtjriLRClOdA3a0OhEokGkaoEOJsSFHuVXmSAC1
Request Date: Oct. 22, 2007 12:50:01 PDT
Done
Remove

Make a note of the API username, API password, and signature. You will need to include this information whenever you execute a PayPal API operation.

Creating an API Certificate

You must establish credentials to use the PayPal API; however, only create an API certificate if your site requires it—most often, you will want to create an API signature for your credentials.

If you really need a certificate, follow the instructions at https://www.paypal.com/IntegrationCenter/ic_api-certificate.html.

NOTE: The certificate you use for API credentials is not the same as an SSL certificate for your website; they are separate entities and not related to each other.

Creating an NVP Request

The Name-Value Pair request format specifies the API operation to perform, credentials that authorize PayPal to access your account, and fields that specify additional information to be used in the request.

Specifying the PayPal API Operation

For the NVP version of the PayPal API, you must specify the name of the PayPal API operation to execute and its version in each request.

The following diagram shows the API operation part of an NVP request:

A method specifies the PayPal operation you want to execute. Each method is associated with a version number; together, the method and version defines the exact behavior of the API operation. Typically, the behavior of an API operation does not change between versions; however, you should carefully retest your code whenever you change a version.

To specify a method and version number

1. Choose the PayPal API operation you want to use.

   `METHOD=`*operation*

2. Choose the appropriate version.

   In most cases, you will want to use the latest version of the API operation.

   `VERSION=`*version_number*

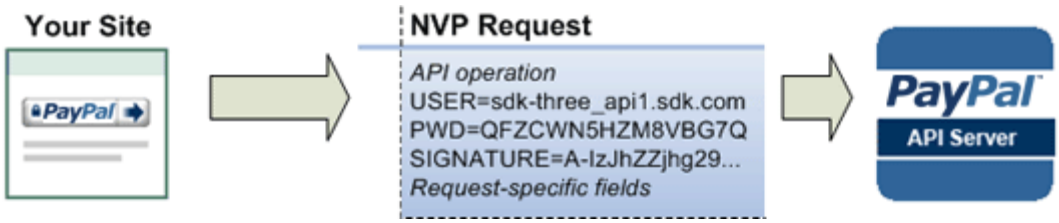
Setting the API operation and version using PHP

```
function PPHttpPost($methodName_, $nvpStr_) {
  ...
  $version = urlencode('52.0');  // NVPRequest for submitting to server
  $nvpreq ="METHOD=$methodName_&VERSION=$version...$nvpStr_";
  ...
}
```

Specifying an API Credential

You must specify API credentials in each request to execute a PayPal API operation.

When you execute a PayPal API operation, you use credentials, such as a signature, to authenticate that you are requesting the API operation. The following diagram shows the API credentials part of an NVP request:



To enable PayPal to authenticate your request

1. Specify the API user name associated with your account.

   USER=*API_username*

2. Specify the password associated with the API user name.

   PWD=*API_password*

3. If you are using an API signature and not an API certificate, specify the API signature associated with the API username.

   SIGNATURE=*API_signature*

Specifying Credentials in PHP

```
$API_UserName = urlencode('my_api_username'); // replace with API username
$API_Password = urlencode('my_api_password'); // replace with password
$API_Signature = urlencode('my_api_signature'); // replace with actual sig
// NVPRequest for submitting to server$nvpreq
  = "...&PWD=$API_Password&USER=$API_UserName&SIGNATURE=$API_Signature...";
```

URL Encoding

All requests to execute PayPal API operations sent via HTTP must be URL encoded.

The PayPal NVP API uses the HTTP protocol to send requests and receive responses from a PayPal API server. You must encode all data sent using the HTTP protocol because data that is not encoded could be misinterpreted as part of the HTTP protocol instead of part of the request. Most programming languages provide a way to encode strings in this way. You should consistently URL encode the complete API request; otherwise, you may find that unanticipated data causes an error.

NOTE: An HTTP form is automatically URL encoded by most browsers.

List Syntax for Name-Value Pairs

The PayPal API uses a special syntax for NVP fields defined as lists.

The NVP interface to the PayPal API requires a unique name for each field. In the API, lists are prefixed by `L_`. To identify an element within the list, use the offset from the beginning of the list, starting with 0 as the first element. For example, `L_DESC0` is the first line of a description, `L_DESC1`, is the second line, and so on.

Executing NVP API Operations

You execute an PayPal NVP API operation by submitting an HTTP POST request to a PayPal API server.

Specifying a PayPal Server

You execute a PayPal API operation by submitting the request to a PayPal API server.

To execute a PayPal NVP API operation, submit your complete request to one of the following end points:

Server end point	Description
`https://api-3t.sandbox.paypal.com/nvp`	Sandbox server for use with API signatures; use for testing your API
`https://api-3t.paypal.com/nvp`	PayPal “live” production server for use with API signatures
`https://api.sandbox.paypal.com/nvp`	Sandbox server for use with API certificates; use for testing your API
`https://api.paypal.com/nvp`	PayPal “live” production server for use with API certificates

NOTE: You must use different API credentials for each kind of server. Typically, you obtain API credentials when you test in the Sandbox and then obtain another set of credentials for the production server. You must change each API request to use the new credentials when you go live.

Logging API Operations

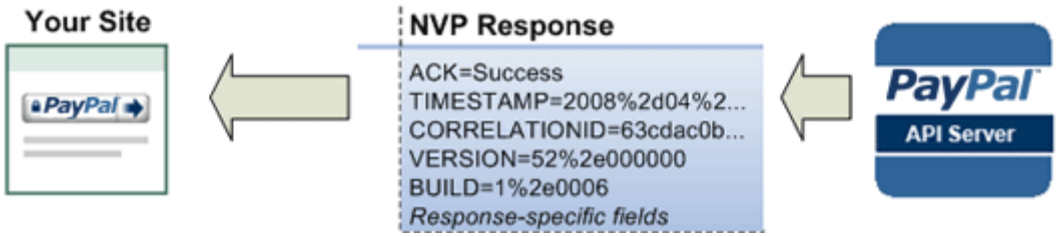
You should log basic information about each PayPal API operation you execute.

All responses to PayPal API operations contain information that may be useful for debugging purposes. You should log the Correlation ID, which identifies the API operation to PayPal, and response-specific information, such as the transaction ID, which you can use to review a transaction on the PayPal website or through the API. You can log other information that may be useful, such as the timestamp. You could implement a scheme that logs the entire request and response in a “verbose” mode; however, you should never log the password from a request.

Responding to an NVP Response

The Name-Value Pair response consists of the answer to the request as well as common fields that identify the API operation and how it was executed.

The following diagram shows fields in the response to a PayPal NVP API operation:



Common Response Fields

The PayPal API always returns common fields in addition to fields that are specific to the requested PayPal API operation.

A PayPal API response includes the following fields:

Field	Description
ACK	Acknowledgement status, which is one of the following values: • `Success` indicates a successful operation • `SuccessWithWarning` indicates a successful operation; however, there are messages returned in the response that you should examine • `Failure` indicates that the operation failed; the response will also contain one or more error message explaining the failure. • `FailureWithWarning` indicates that the operation failed and that there are messages returned in the response that you should examine
CORRELATIONID	Correlation ID, which uniquely identifies the transaction to PayPal
TIMESTAMP	The date and time that the requested API operation was performed
VERSION	The version of the API
BUILD	The sub-version of the API

URL Decoding

All responses to HTTP POST operations used by the PayPal NVP API must be decoded.

The PayPal NVP API uses the HTTP protocol to send requests and receive responses from a PayPal API server. You must decode all data returned using the HTTP protocol so that it can be displayed properly. Most programming languages provide a way to decode strings.

NOTE: Most browsers decode responses to HTTP requests automatically.

4 Implementing the Simplest Express Checkout Integration

The simplest Express Checkout integration requires the following PayPal API operations: `SetExpressCheckout`, `DoExpressCheckoutPayment`, and optionally, `GetExpressCheckoutDetails`.

- Setting Up the Express Checkout Transaction
- Obtaining Express Checkout Transaction Details
- Completing the Express Checkout Transaction

Setting Up the Express Checkout Transaction

To set up an Express Checkout transaction, you must invoke the `SetExpressCheckout` API to provide sufficient information to initiate the payment flow and redirect to PayPal if the operation was successful.

This example assumes that you have set up the mechanism you will use to communicate with the PayPal server and have a PayPal business account with API credentials. It also assumes that the payment action is a final sale.

When you set up an Express Checkout transaction, you specify values in the `SetExpressCheckout` request and then call the API. The values you specify control the PayPal page flow and the options available to you and your buyers. You should start by setting up a standard Express Checkout transaction, which can be modified to include additional options.

To set up the simplest standard Express Checkout transaction

1. Specify the amount of the transaction; include the currency if it is not in US dollars.

   Specify the total amount of the transaction if it is known; otherwise, specify the subtotal. Regardless of the specified currency, the format must have decimal point with exactly two digits to the right and an optional thousands separator to the left, which must be a comma; for example, EUR 2.000,00 must be specified as 2000.00 or 2,000.00. The specified amount cannot exceed USD $10,000.00, regardless of the currency used.

   AMT=*amount*
   CURRENCYCODE=*currencyID*

2. Specify the return URL.

   The return URL is the page to which PayPal redirects your buyer's browser after the buyer logs into PayPal and approves the payment. Typically, this is a secure page (`https://...`) on your site.

NOTE: You can use the return URL to piggyback parameters between pages on your site. For example, you can set your Return URL to specify additional parameters using the `https://www.`*yourcompany*`.com/`*page*`.html?`*param=value*... syntax. The parameters become available as request parameters on the page specified by the Return URL.

```
RETURNURL=return_url
```

3. Specify the cancel URL.

   The cancel URL is the page to which PayPal redirects your buyer's browser if the buyer does not approve the payment. Typically, this is the secure page (`https://...`) on your site from which you redirected the buyer to PayPal.

   NOTE: You can pass `SetExpressCheckout` request values as parameters in your URL to have the values available, if necessary, after PayPal redirects to your URL.

```
CANCELURL=cancel_url
```

4. Specify the payment action.

   Although the default payment action is a `Sale`, it is a best practice to explicitly specify the payment action as one of the following values:

```
PAYMENTACTION=Sale
PAYMENTACTION=Authorization
PAYMENTACTION=Order
```

5. Execute the `SetExpressCheckout` API operation to set up the Express Checkout transaction.
6. Test that the response to the `SetExpressCheckout` API operation was successful.
7. If calling the `SetExpressCheckout` API was successful, redirect the buyer's browser to PayPal and execute the `_express-checkout` command using the token returned in the `SetExpressCheckout` response.

NOTE: The following example uses the PayPal Sandbox server:

```
https://www.sandbox.paypal.com/webscr
        ?cmd=_express-checkout&token=tokenValue
        &AMT=amount
        &CURRENCYCODE=currencyID
        &RETURNURL=return_url
        &CANCELURL=cancel_url
```

Obtaining Express Checkout Transaction Details

To obtain details about an Express Checkout transaction, you can invoke the `GetExpressCheckoutDetails` API operation.

This example assumes that PayPal redirects to your buyer's browser with a valid token after the buyer reviews the transaction on PayPal.

Although you are not required to invoke the `GetExpressCheckoutDetails` API operation, most Express Checkout implementations take this action to obtain information about the buyer. You invoke the `GetExpressCheckoutDetails` API operation from the page specified by return URL, which you set in your call to the `SetExpressCheckout` API. Typically, you invoke this operation as soon as the redirect occurs and use the information in the response to populate your review page.

To obtain a buyer's shipping address and Payer ID

1. Specify the token returned by PayPal when it redirects the buyer's browser to your site.

   PayPal returns the token to use in the `token` HTTP request parameter when redirecting to the URL you specified in your call to the `SetExpressCheckout` API.

   `TOKEN=`*`tokenValue`*

2. Execute the `GetExpressCheckoutDetails` API to obtain information about the buyer.
3. Access the fields in the `GetExpressCheckoutDetails` API response.

   NOTE: Only populated fields are returned in the response.

Completing the Express Checkout Transaction

To complete an Express Checkout transaction, you must invoke the `DoExpressCheckoutPayment` API operation.

This example assumes that PayPal redirects your buyer's browser to your website with a valid token after you call the `SetExpressCheckout` API. Optionally, you may call the `GetExpressCheckoutDetails` API before calling the `DoExpressCheckoutPayment` API.

In the simplest case, you set the total amount of the order when you call the `SetExpressCheckout` API. However, you can change the amount before calling the `DoExpressCheckoutPayment` API if you did not know the total amount when you called the `SetExpressCheckout` API.

This example assumes the simplest case, in which the total amount was specified in the return URL when calling the `SetExpressCheckout` API. Although you can specify additional options, this example does not use any additional options.

To execute an Express Checkout transaction

1. Specify the token returned by PayPal when it redirects the buyer's browser to your site.

   PayPal returns the token to use in the `token` HTTP request parameter when redirecting to the URL you specified in your call to the `SetExpressCheckout` API.

   ```
   TOKEN=tokenValue
   ```

2. Specify the Payer ID returned by PayPal when it redirects the buyer's browser to your site.

   PayPal returns the Payer ID to use in the `token` HTTP request parameter when redirecting to the URL you specified in your call to the `SetExpressCheckout` API. Optionally, you can obtain the Payer ID by calling the `GetExpressCheckoutDetails` API.

   ```
   PAYERID=id
   ```

3. Specify the amount of the order including shipping, handling, and tax; include the currency if it is not in US dollars.

   Regardless of the specified currency, the format must have decimal point with exactly two digits to the right and an optional thousands separator to the left, which must be a comma; for example, EUR 2.000,00 must be specified as 2000.00 or 2,000.00. The specified amount cannot exceed USD $10,000.00, regardless of the currency used.

   ```
   AMT=amount
   CURRENCYCODE=currencyID
   ```

4. Specify the payment action.

   Although the default payment action is a `Sale`, it is a best practice to explicitly specify the payment action as one of the following values:

   ```
   PAYMENTACTION=Sale
   PAYMENTACTION=Authorization
   PAYMENTACTION=Order
   ```

5. Execute the `DoExpressCheckoutPayment` API to complete the Express Checkout transaction.

6. Examine the values returned by the API if the transaction completed successfully.

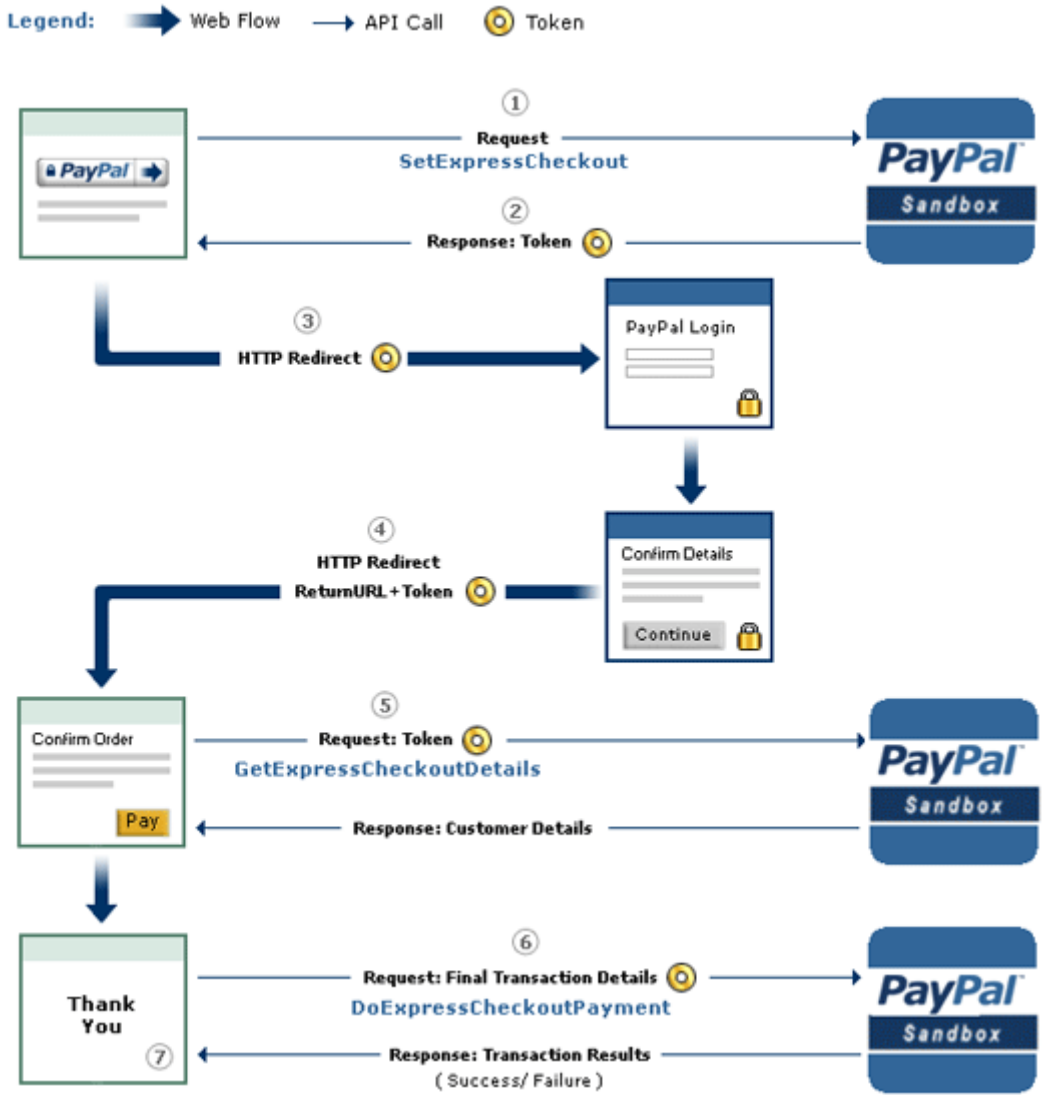
5 Testing an Express Checkout Integration

You can test your Express Checkout integration in the Sandbox.

This example shows how to simulate your web pages using HTTP forms and supplying the values for API operations from these forms. You can use this strategy for your initial testing; however, for more complete testing, you will want to replace these forms with your web pages containing actual code.

The following diagram shows the Express Checkout execution flow, which uses the Sandbox as the API server. The pages on the left represent your site.

Express Checkout Execution Flow



The following steps match the circled numbers in the diagram. Perform the actions in each step to test Express Checkout.

1. Invoke a form on your site that calls the `SetExpressCheckout` API on the Sandbox.

To invoke the API, set form fields whose names match the NVP names of the fields you want to set, specify their corresponding values, and then post the form to a PayPal Sandbox server, such as `https://api-3t.sandbox.paypal.com/nvp`, as shown in the following example:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
   <input type=hidden name=USER value= API_username>
   <input type=hidden name=PWD value= API_password>
   <input type=hidden name=SIGNATURE value= API_signature>
   <input type=hidden name=VERSION value=52.0>
   <input type=hidden name=PAYMENTACTION value=Sale>
   <input name=AMT value=19.95>
   <input type=hidden name=RETURNURL
      value=http://www.YourReturnURL.com>
   <input type=hidden name=CANCELURL
      value=http://www.YourCancelURL.com>
   <input type=submit name=METHOD value=SetExpressCheckout>
</form>
```

NOTE: The API username is a Sandbox business test account for which a signature exists. See the Test Certificates tab of the Sandbox to obtain a signature. If you are not using a signature, you must use a different Sandbox server.

2. Review the response string from the `SetExpressCheckout` API operation.

   PayPal responds with a message, such as the one shown below. Note the status, which should include `ACK` set to `Success`, and a token that is used in subsequent steps.

```
TIMESTAMP=2007%2d04%2d05T23%3a23%3a07Z
&CORRELATIONID=63cdac0b67b50
&ACK=Success
&VERSION=52%2e000000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
```

3. If the operation was successful, use the token and redirect your browser to the Sandbox to log in, as follows:

```
https://www.sandbox.paypal.com/cgi-bin/webscr?
cmd=_express-checkout
&token=EC-1NK66318YB717835M
```

   NOTE: The `token` in the command line is an HTTP request parameter and not the NVP field; this parameter name must be lowercase.

   You may need to replace hexadecimal codes with ASCII codes; for example, you may need to replace `%2d` in the token with a hyphen (`-`).

   You must log in to `https://developer.paypal.com` before you log in to a Sandbox test account. You then log in to the test account that represents the buyer, not the API_username business test account that represents you as the merchant.

4. After logging into the buyer test account, confirm the details.

   When you confirm, the Sandbox redirects your browser to the return URL you specified when invoking the `SetExpressCheckout` API operation, as in the following example:

```
http://www.YourReturnURL.com/
                    ?token=EC-1NK66318YB717835M&PayerID=7AKUSARZ7SAT8
```

5. Invoke a form on your site that calls the `GetExpressCheckoutDetails` API operation on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp
   <input type=hidden name=USER value=API_username>
   <input type=hidden name=PWD value=API_password>
   <input type=hidden name=SIGNATURE value=API_signature>
   <input type=hidden name=VERSION value=52.0>
   <input name=TOKEN value=EC-1NK66318YB717835M>
   <input type=submit name=METHOD value=GetExpressCheckoutDetails>
</form>
```

If the operation was successful, the `GetExpressCheckoutDetails` API returns information about the payer, such as the following information:

```
TIMESTAMP=2007%2d04%2d05T23%3a44%3a11Z
&CORRELATIONID=6b174e9bac3b3   &ACK=Success
&VERSION=52%2e000000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&EMAIL=jsmith01@example.com
&PAYERID=7AKUSARZ7SAT8
&PAYERSTATUS=verified
&FIRSTNAME=...
&LASTNAME=...
&COUNTRYCODE=US
&BUSINESS=...
&SHIPTONAME=...
&SHIPTOSTREET=...
&SHIPTOCITY=...
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOCOUNTRYNAME=United%20States
&SHIPTOZIP=94666
&ADDRESSID=...
&ADDRESSSTATUS=Confirmed
```

6. Invoke a form on your site that invokes the `DoExpressCheckoutPayment` API operation on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
   <input type=hidden name=USER value=API_username>
   <input type=hidden name=PWD value=API_password>
   <input type=hidden name=SIGNATURE value=API_signature>
   <input type=hidden name=VERSION value=52.0>
   <input type=hidden name=PAYMENTACTION value=Sale>
   <input type=hidden name=PAYERID value=7AKUSARZ7SAT8>
   <input type=hidden name=TOKEN value= EC%2d1NK66318YB717835M>
   <input type=hidden name=AMT value= 19.95>
   <input type=submit name=METHOD value=DoExpressCheckoutPayment>
</form>
```

7. Review the response string from the `DoExpressCheckoutPayment` API operation.

   If the operation was successful, the response should include `ACK` set to `Success`, as follows:

```
TIMESTAMP=2007%2d04%2d05T23%3a30%3a16Z
&CORRELATIONID=333fb808bb23   &ACK=Success
&VERSION=52%2e000000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&TRANSACTIONID=043144440L487742J
&TRANSACTIONTYPE=expresscheckout
&PAYMENTTYPE=instant
&ORDERTIME=2007%2d04%2d05T23%3a30%3a14Z
&AMT=19%2e95
&CURRENCYCODE=USD
&TAXAMT=0%2e00
&PAYMENTSTATUS=Completed
&PENDINGREASON=None
&REASONCODE=None
&FEEAMT=0%2e43
```

5 Testing an Express Checkout Integration

6 Customizing Express Checkout

You can specify options in Express Checkout API requests that change the appearance, behavior, and flow of the checkout process.

- PayPal Review Page Order Details
- PayPal Page Style
- Changing the Locale
- Handling Shipping Addresses
- Automatically Filling Out the PayPal Login Page
- User Confirms Order on PayPal

PayPal Review Page Order Details

NOTE: The `DoExpressCheckoutPayment` request includes the same order details and eBay auction incentive parameters as `SetExpressCheckout`. It is strongly recommended that you submit the same parameters in both API calls.

When a buyer logs into PayPal to check out, you can present the buyer with detailed information about each item in the shopping cart (see PayPal Review page with order details):

(1) – Item name. The item name can identify this item to distinguish it from other line items in the order.

(2) – Item number. Each item can be further identified by an item number. If the item is an eBay auction item, it is recommended that you provide the eBay item number in this field.

(3) – Item description. This field identifies which of several items the buyer is purchasing. For example, you may be offering an item in different sizes. Knowing the size helps the buyer decide whether the one they selected was appropriate. If the item is an eBay auction item, it is recommended that you provide the phrase “eBay item” in this field.

(4) – Item unit price. This field specifies exactly how much one unit of the item costs. It can be a positive or negative value but not zero.

(5) – Item unit quantity. This field identifies the number of units the buyer is ordering.

PayPal calculates the value in the Amount **(6)** column as the product of line-item unit price and line-item unit quantity.

You can also show other detailed information about the order:

(7) – Item total and tax. These are the respective total of all items in the order and the tax.

(8) – Shipping and handling. PayPal sums the shipping and handling amounts in this field. (You determine actual shipping and handling amounts.)

(9) – Shipping discount. If the buyer is receiving a discount on shipping, the value appears as a credit in this field.

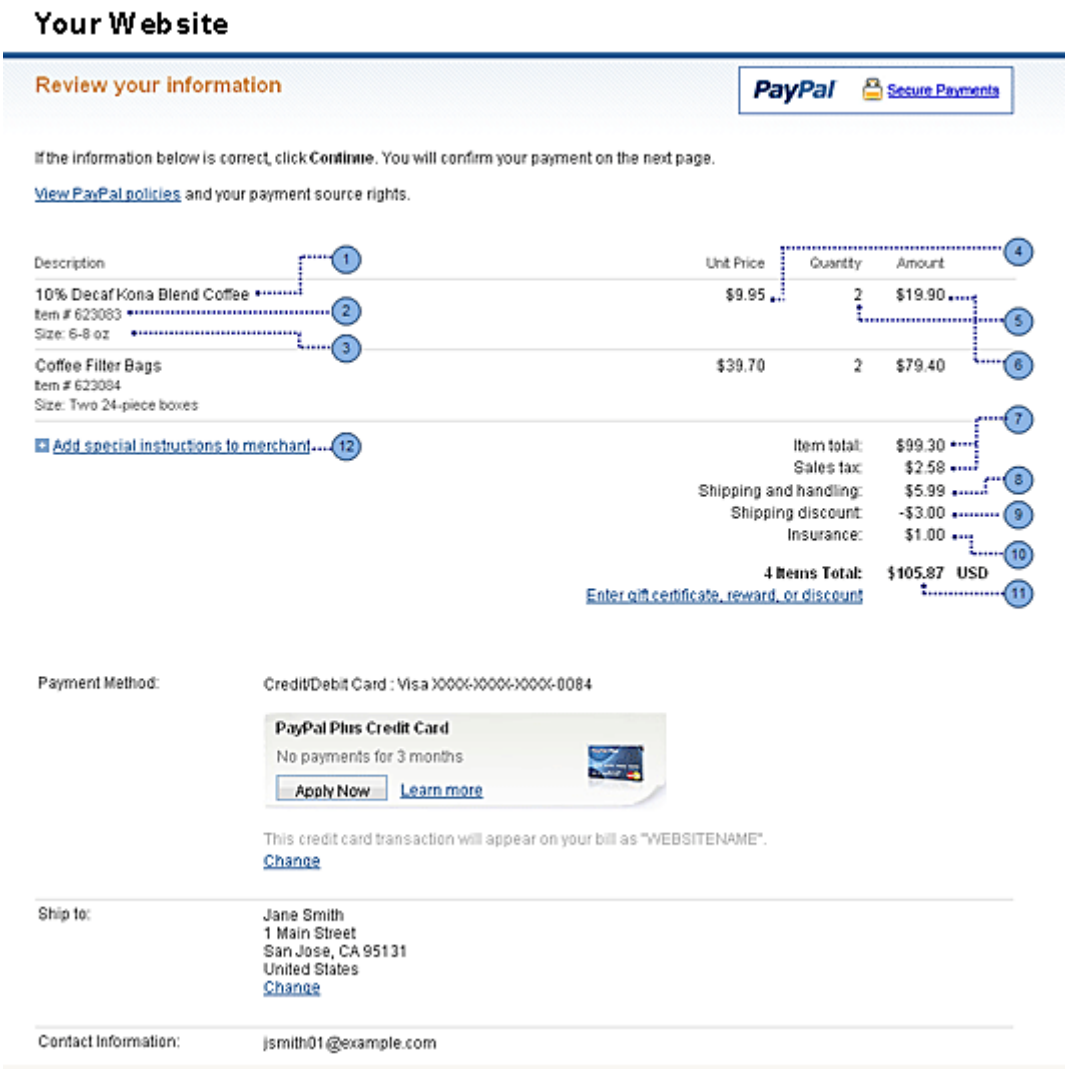
(10) – Insurance. If there is insurance on shipping, the insurance fee is shown in this field.

(11) – Total. This is the total of the order, including shipping, handling, tax, and other price adjustment-related items.

NOTE: The 'Enter gift certificate, reward, or discount' link enables the buyer to redeem certificates, rewards, or discounts PayPal issues. The link does not enable the buyer to redeem incentives that you issue.

(12) – Note. You can allow the buyer to send you special instructions about the order in a text box that appears when the buyer clicks this link.

PayPal Review page with order details



Special Instructions to Merchant

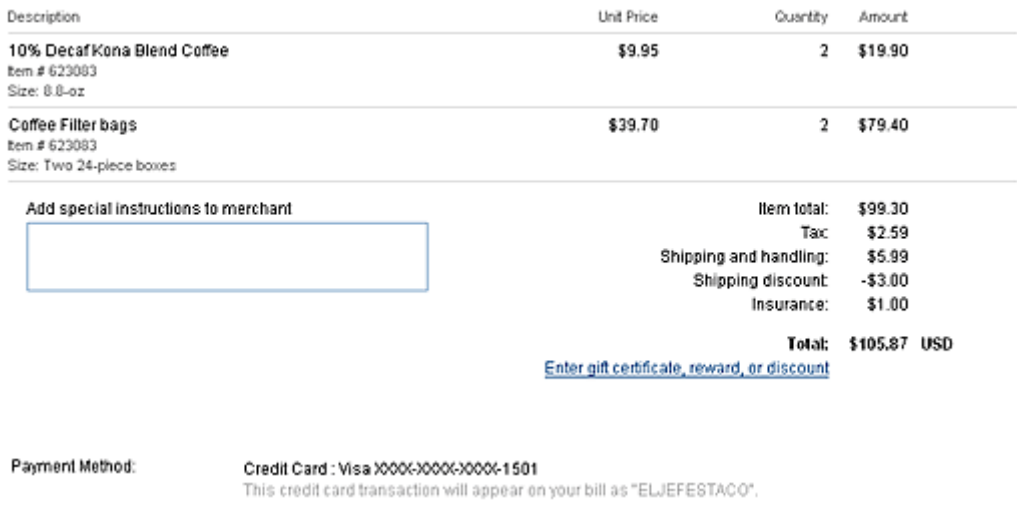
You can allow the buyer to send you special instructions about an order. This feature is especially helpful to buyers who want to customize merchandise. A buyer also might want to tell you to ship their order at a later date because they are out of the country.

NOTE: Users of this feature should be sure to read the instructions the buyer sends.

This feature appears as the link just below the item description on the PayPal Review page **(12)**. When the buyer clicks the Add special instructions to merchant link, a text box opens, as shown in Special Instructions text box. After the buyer enters special instructions and clicks

Continue, the instructions are returned in the responses to `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

Special Instructions text box



Integrating Order Details into the Express Checkout Flow

To integrate order details into the checkout flow, pass the Express Checkout parameters described in SetExpressCheckout Order Details Parameters to `SetExpressCheckout`.

SetExpressCheckout Order Details Parameters

NVP Field	SOAP Field	Description and Comments
L_NAMEn	Name	Item name.
L_NUMBERn	Number	Item number.
L_DESCn	Description	Item description. The DESC (NVP) and OrderDescription (SOAP) fields still exist for backwards compatibility. However, L_DESCn and Description enable you to provide a more precise description for each different item purchased such as hiking boots or cooking utensils rather than one general purpose description such as camping supplies.
L_AMTn	Amount	Item unit price. PayPal calculates the product of the item unit price and item unit quantity (below) in the Amount column of the PayPal Review page, as shown in PayPal Review page with order details. The item unit price can be a positive or a negative value but not 0. You may provide a negative value to reflect a discount on an order, for example.
L_QTYn	Quantity	Item unit quantity.
ITEMAMT	ItemTotal	Sum of cost of all items in this order.
TAXAMT	TaxTotal	Sum of tax for all items in this order.
SHIPPINGAMT	ShippingTotal	Total shipping cost for this order. PayPal calculates the sum of the shipping cost and the handling cost (below) in the PayPal Review page, as shown at **(8)** in PayPal Review page with order details. Pass in a shipping amount that is reasonably accurate, because you may need to change it later on your final checkout page. You might assume, for example, the buyer lives in California because your business is in California.
HANDLINGAMT	HandlingTotal	Total handling cost for this order. PayPal calculates the sum of the handling cost and the shipping cost (above), as shown at **(8)** in PayPal Review page with order details.
SHIPDISCAMT	ShippingDiscount	Shipping discount for this order. You specify this value as a negative number.
INSURANCEAMT	InsuranceAmount	Total shipping insurance cost for this order.
AMT	OrderTotal	Total of order, including shipping, handling, tax, and any other billing adjustments such as a credit due.
ALLOWNOTE	AllowNote	Provide a value of 1 to indicate that the buyer may enter a note to you on the PayPal Review page during checkout. See Special Instructions text box.

The following example shows how to set the above parameters in the call to SetExpressCheckout.

Request Parameters:

[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTACTION=Sale
&NAME=J Smith
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&L_NAME0=10% Decaf Kona Blend Coffee
&L_NUMBER0=623083
&L_DESC0=Size: 8.8-oz
&L_AMT0=9.95
&L_QTY0=2
&L_NAME1=Coffee Filter bags
&L_NUMBER1=623084
&L_DESC1=Size: Two 24-piece boxes
&L_AMT1=39.70
&L_QTY1=2
&ITEMAMT=99.30
&TAXAMT=2.58
&SHIPPINGAMT=3.00
&HANDLINGAMT=2.99
&SHIPDISCAMT=-3.00
&INSURANCEAMT=1.00
&AMT=105.87
&CURRENCYCODE=USD
&ALLOWNOTE=1

Response Parameters

[successResponseFields]
&TOKEN=EC-17C76533PL706494P

You can provide values for any subset of the Express Checkout parameters shown in SetExpressCheckout Order Details Parameters in the call to SetExpressCheckout.

If you pass the generic order description parameter (DESC) and any one line-item parameter listed below, both parameter values display on the Review page. If you pass the generic order description parameter (DESC) along with any two of the following line-item parameters, the order description value does not display.

- L_NAMEn
- L_NUMBERn
- L_DESCn

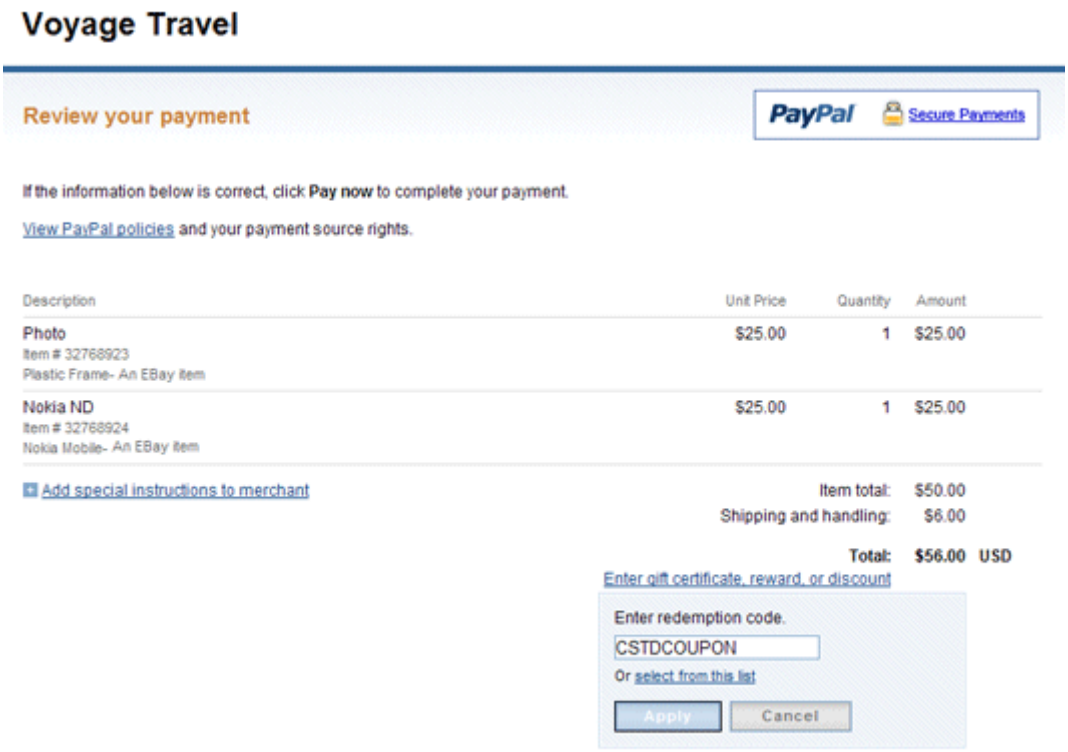
If you pass in unit price information (L_AMTn) without passing in the unit quantity (L_QTYn), the unit price will not display. To show both values, you must pass in values for both parameters. You can pass in a value of 1 even if the item purchase is uncountable.

eBay-Issued Incentives

If the buyer's checkout includes eBay auction items, you must set additional fields in the call to `SetExpressCheckout` to enable buyers to choose from eBay-issued incentives such as eBay gift certificates, coupons, vouchers, and gift cards that may apply to their order.

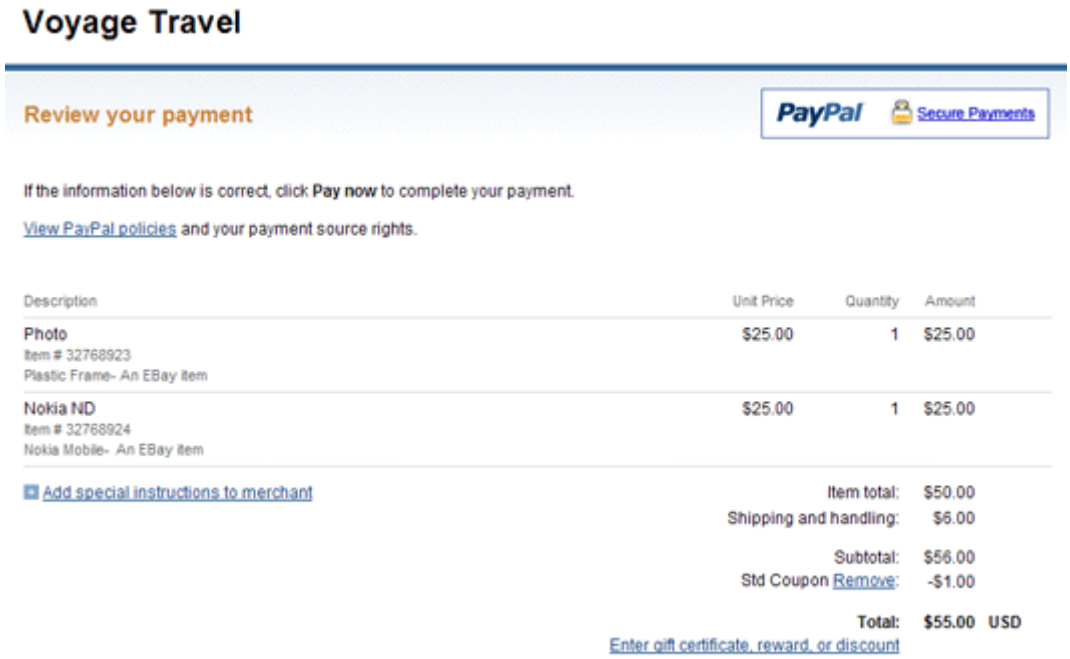
To enable the display of eBay incentives on the PayPal Review page, you also must provide line item detail information for each eBay item. When the buyer clicks the Enter gift certificate, reward, or discount link, a dialog displays in which the buyer can either enter a redemption code or select from a list of all discounts applicable to eBay items in the cart display as shown below. If multiple items display, the buyer can select from the discounts they want applied to their order.

Dialog for entering redemption code or choosing discounts



After entering a redemption code or selecting the incentives, the PayPal Review page reflects changes to the order. Incentives appear as credits and a new Total is calculated as shown below.

New Total after applying incentives



Integrating eBay Incentives into the Express Checkout Flow

To integrate eBay incentives into the checkout flow:

1. Pass the following parameters and settings in the call to `SetExpressCheckout`.
   - `PAYMENTACTION` must be `Sale`
   - `CHANNELTYPE` must be `eBayItem`

   NOTE: For eBay auctions, you cannot set `PAYMENTACTION` to `Authorization` or `Order`. You must set `PAYMENTACTION` to `Sale`.

2. For each eBay line item, do one of the following (mutually exclusive choices) in the call to `SetExpressCheckout` and in the call to `DoExpressCheckoutPayment`:

   NOTE: `DoExpressCheckoutPayment` includes the same order details and eBay auction parameters as `SetExpressCheckout`. It is strongly suggested that, if you pass line-item detail parameter data in `SetExpressCheckout`, you pass the same parameter data in the call to `DoExpressCheckoutPayment`. This ensures that order details display on the PayPal Review page and payment is properly reflected in My eBay and in PayPal transaction details.

   - For each individual eBay entry, pass the values returned by eBay when setting up the auction item descriptions in the parameters below:

     `L_EBAYITEMNUMBERn` and `L_EBAYITEMAUCTIONTXNIDn`

– For multiple eBay items offered as a single order, pass the value returned by eBay when setting up the auction order description in the parameter below:

  `L_EBAYITEMORDERIDn`

3. It is recommended that you pass the string “eBay item” in `L_DESCn` and the eBay Item Number in `L_NUMBERn`.

4. If you host the final payment confirmation page on your website by setting `useraction=continue` in the redirect, you must show the discount amount resulting from eBay incentives on that page. You can get the discount amount from the `PAYPALADJUSTMENT` value returned in the response to `GetExpressCheckoutDetails`.

NOTE: eBay maintains detailed information about each item sold in eBay auctions. If you set the value for `SHIPPINGAMT` in the call to `SetExpressCheckout`, for example, you do not need to be concerned about the shipping cost for individual items. eBay calculates incentives based on its database, not from the values you pass in the `SetExpressCheckout` call.

The following `SetExpressCheckout` request example passes order details and eBay incentives for two eBay auction items:

Request Parameters:

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=http://coffee2go.com
&CANCELURL=http://cancel.com
&PAYMENTACTION=Sale
&EMAIL=jsmith01@example.com
&NAME=J Smith
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&L_NAME0=Photo
&L_NUMBER0=32768923
&L_DESC0=Plastic Frame- An EBay item
&L_AMT0=25.00
&L_QTY0=1
&L_EBAYITEMNUMBER0=32768923
&L_EBAYITEMAUCTIONTXNID0=0
&L_NAME1=Nokia ND
&L_NUMBER1=32768924
&L_DESC1=Nokia Mobile- An EBay item
&L_AMT1=25.00
&L_QTY1=1
&L_EBAYITEMNUMBER1=32768924
&L_EBAYITEMAUCTIONTXNID1=7225687004
&CHANNELTYPE=eBayItem
&ITEMAMT=50.00
&SHIPPINGAMT=3.00
&HANDLINGAMT=3.00
&AMT=56.00
&CURRENCYCODE=USD
&ALLOWNOTE=1
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-2HX34015EC629990M
```

PayPal Page Style

You can change the overall appearance of the PayPal review page by defining a custom page style or by customizing individual page style characteristics.

You define a custom page style in the PayPal Profile and then pass the resulting page style name in the call to `SetExpressCheckout`. Typically you customize individual page style characteristics in the PayPal Profile as well. However, you can also call `SetExpressCheckout` and pass in individual page characteristics dynamically.

Custom Page Style

When your buyer logs into PayPal to check out, you can make the PayPal pages the buyer sees appear to have a similar look and feel to those on your website. You can customize any of these page characteristics and save the results as a Page Style Name: Up to three unique Page Style Names can be defined in your account. You can specify the following items:

- Header image
- Header border color
- Header background color
- Page background color

For instructions on how to customize page styles and create Page Style Names, see the Merchant Setup and Administration Guide.

To set a custom page style in a call to `SetExpressCheckout`:

1. Include the optional `PAGESTYLE` parameter in the call to `SetExpressCheckout`.
2. Set `PAGESTYLE` to the Page Style Name you defined in your account.

The following example sets `PAGESTYLE` to the Page Style Name:

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00&CURRENCYCODE=USD
&PAYMENTACTION=Sale&
&PAGESTYLE=TestMerchant
&SHIPTOSTREET=1 Main St
SHIPTOSTREET2=
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706494P
```

Individual Page Style Characteristics

Typically, you create a custom page style for the PayPal pages using the Custom Payment Pages option from the PayPal Profile tab. In cases where you do not want to use the Profile option, you can specify any of the individual page style characteristics:

- Header image

- Header border color
- Header background color
- Page background color

To define a header image such as your company logo:

1. Create a header image up to 750 pixels wide by 90 pixels high, and save it in a valid graphics format such as .gif, .jpg, or .png.
2. Store the URL to the image on a secure (https) server so your buyer's web browser does not display a message that the payment contains insecure items.
3. Assign the URL to the `HDRIMG` parameter in the call to `SetExpressCheckout`.

The following example sets `HDRIMG` to a custom header image.

Request Parameters

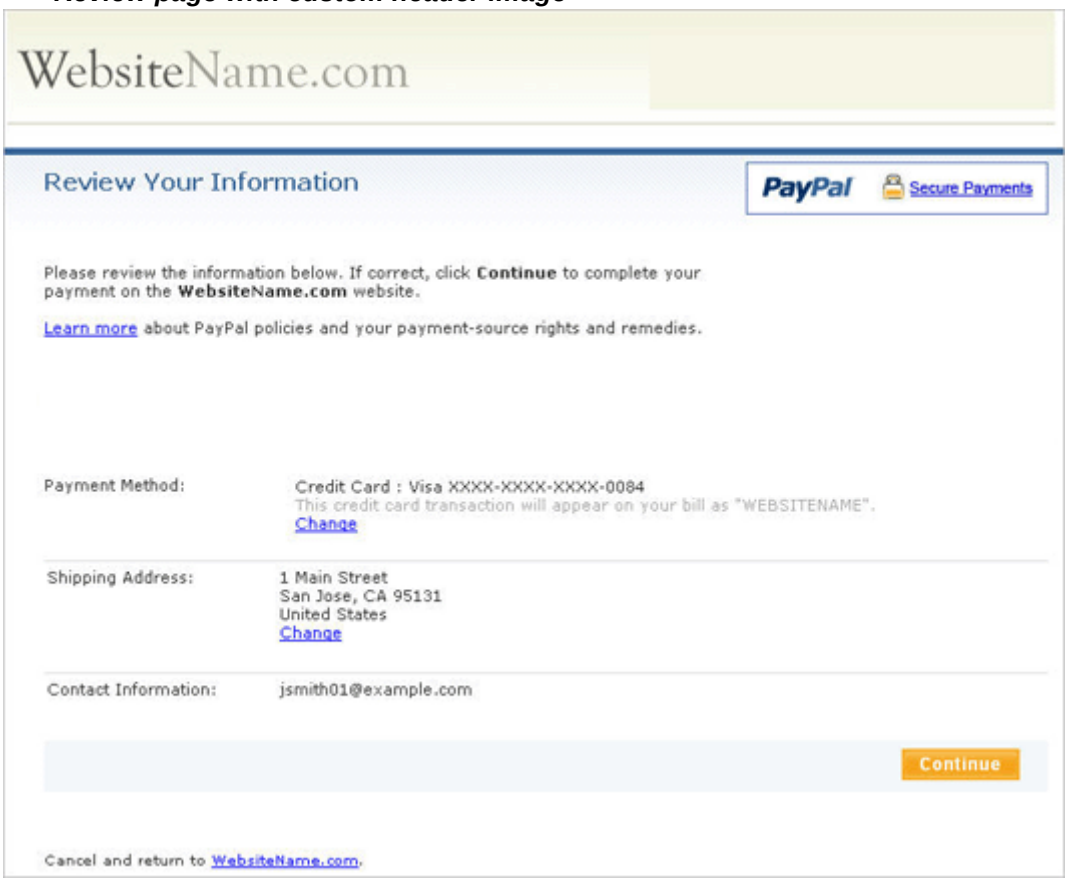
```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&MAXAMT=
&PAYMENTACTION=Sale
&HDRIMG=https://www.mayadeviimports.com/ebay/websitename-logo.gif
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706494P
```

The figure below shows the review page with a custom header image set in the `SetExpressCheckout` request.

Review page with custom header image



The following example sets colors for the header background (HDRBACKCOLOR) and header border (HDRBORDERCOLOR).

Request Parameters

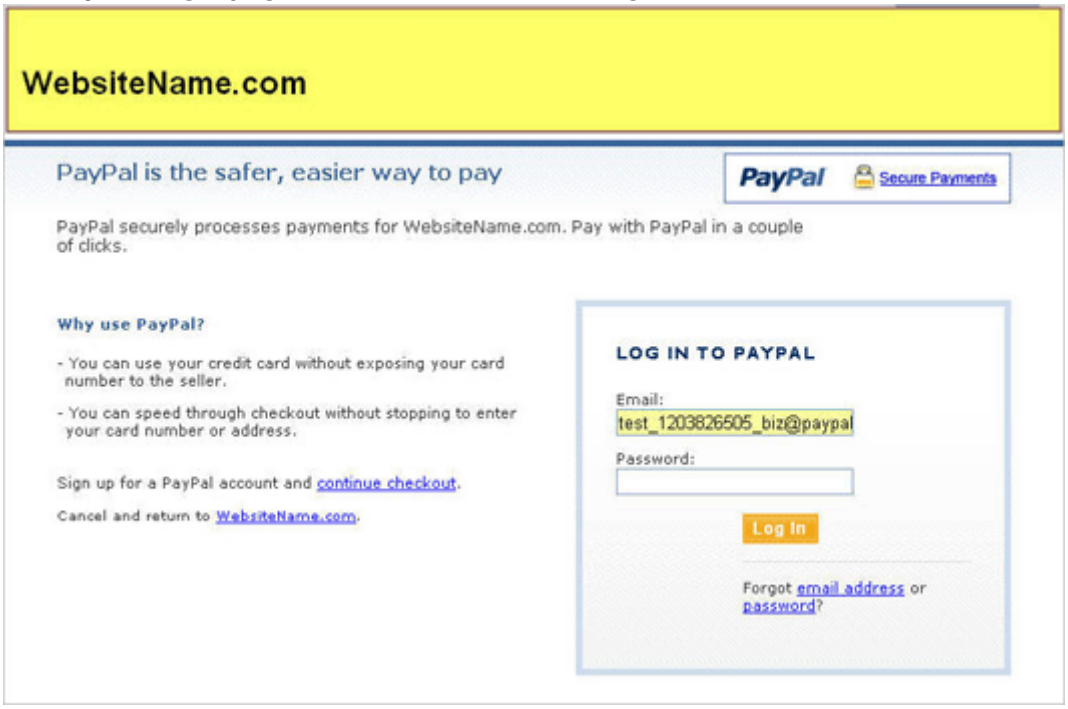
[requiredSecurityParameters]
&**METHOD=SetExpressCheckout**
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&MAXAMT=
&PAYMENTACTION=Sale
&**HDRBACKCOLOR=FFFF66**
&**HDRBORDERCOLOR=996666**
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444

Response Parameters

[successResponseFields]
`&TOKEN=EC-17C76533PL706494P`

The figure below shows the PayPal Login page with `HDRBACKCOLOR` and `HDRBORDERCOLOR` set to custom colors.

PayPal Login page with custom header background and border colors


WebsiteName.com
PayPal is the safer, easier way to pay
PayPal
Secure Payments
PayPal securely processes payments for WebsiteName.com. Pay with PayPal in a couple of clicks.
Why use PayPal?
- You can use your credit card without exposing your card number to the seller.
- You can speed through checkout without stopping to enter your card number or address.
Sign up for a PayPal account and continue checkout.
Cancel and return to WebsiteName.com.
LOG IN TO PAYPAL
Email:
test_1203826505_biz@paypal
Password:
Log In
Forgot email address or password?

Changing the Locale

You can change the locale of PayPal pages to match the language on your website.

To change the language displayed on the PayPal Login page, set the `LOCALECODE` parameter to one of the following allowable values in the `SetExpressCheckout` call:

- AU
- DE
- FR
- GB
- IT
- ES
- JP
- US

The following example sets LOCALCODE to ES (Spain).

Request Parameters

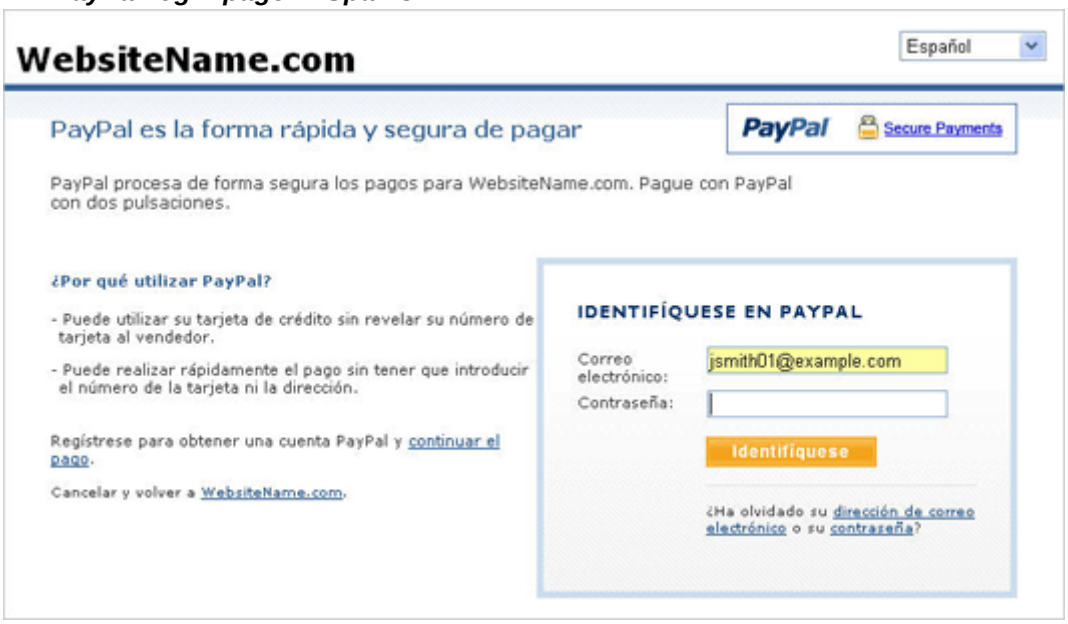
[requiredSecurityParameters]
&**METHOD=SetExpressCheckout**
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&CURRENCYCODE=EUR
PAYMENTACTION=Sale
&**LOCALECODE=ES**
&SHIPTOSTREET=1 Main St
&SHIPTOSTREET2=
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444

Response Parameters

[successResponseFields]
&TOKEN=EC-17C76533PL706494P

The figure below shows the PayPal Login page when the LOCALECODE is set to ES.

PayPal login page in Spanish


WebsiteName.com
Español
PayPal es la forma rápida y segura de pagar
PayPal
Secure Payments
PayPal procesa de forma segura los pagos para WebsiteName.com. Pague con PayPal con dos pulsaciones.
¿Por qué utilizar PayPal?
- Puede utilizar su tarjeta de crédito sin revelar su número de tarjeta al vendedor.
- Puede realizar rápidamente el pago sin tener que introducir el número de la tarjeta ni la dirección.
Regístrese para obtener una cuenta PayPal y continuar el pago.
Cancelar y volver a WebsiteName.com.
IDENTIFÍQUESE EN PAYPAL
Correo electrónico:
jsmith01@example.com
Contraseña:
Identifíquese
¿Ha olvidado su dirección de correo electrónico o su contraseña?

Handling Shipping Addresses

You can specify several shipping address options that affect the PayPal Review page.

In your SetExpressCheckout request, you can specify the following options:

- Require a confirmed address
- Not display the shipping address on the review page
- Display an alternative address on the review page.

Confirmed Address

A *confirmed address* is a shipping address that PayPal has established as belonging to the PayPal account holder. To be protected by PayPal's Seller Protection Policy, you must require the shipping address be a confirmed address.

NOTE: Because many buyers prefer to ship to a non-confirmed address (they may, for example, be shipping a gift to someone), PayPal does not recommend requiring a confirmed address unless you are selling high-risk merchandise. If you prefer confirmed addresses be used, then do not set ADDROVERRIDE, as described in Shipping Address Override.

To require a confirmed address for the shipping address, ensure that the shipping address matches the address on record with PayPal. You can do this through your account profile, as described in the *Merchant Setup and Administration Guide*. The guide is available on the Documentation page linked to the Library tab on Developer Central. Alternately, you can set a flag in the call to SetExpressCheckout as described below:

1. Include the optional REQCONFIRMSHIPPING parameter in the call to SetExpressCheckout.
2. Set REQCONFIRMSHIPPING to 1.

The following example requires the shipping address be a confirmed address.

NOTE: The value of REQCONFIRMSHIPPING overrides the setting in your Merchant Account Profile.

Request Parameters

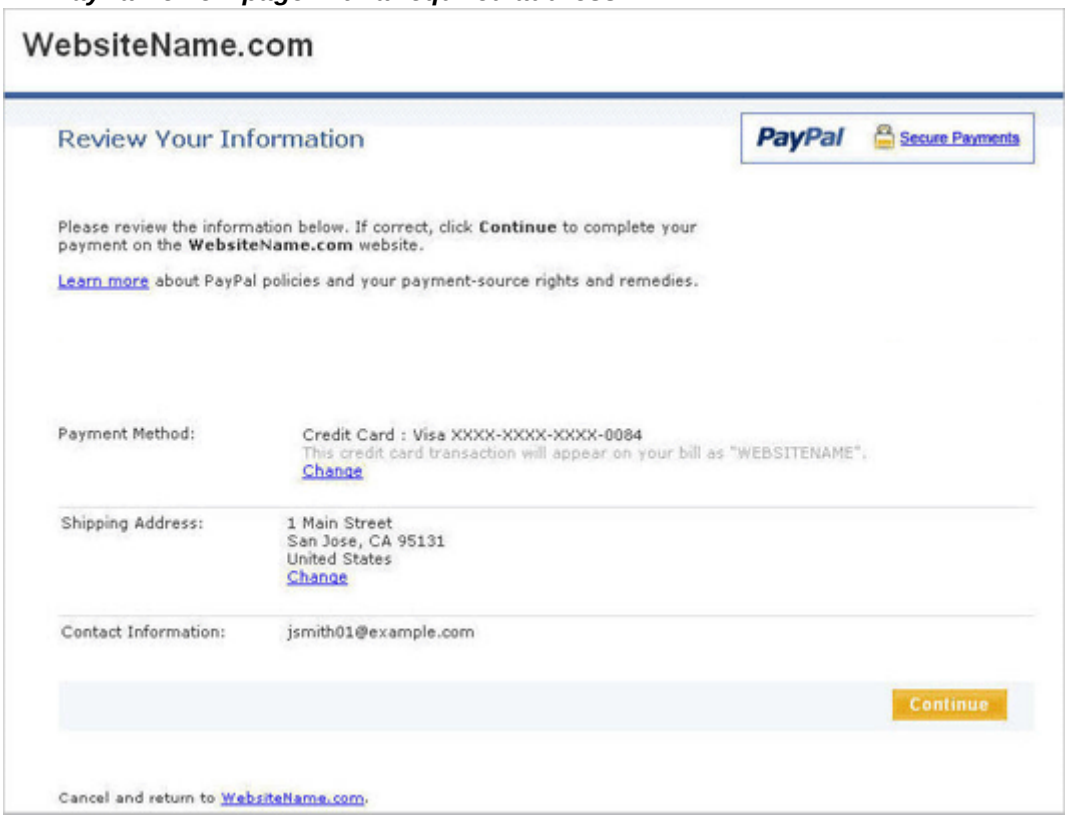
```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&PAYMENTACTION=Sale
&REQCONFIRMSHIPPING=1
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444
```

Response Parameters

[successResponseFields]
`&TOKEN=EC-6UA07551EA393551U`

The figure below shows the PayPal review page when `REQCONFIRMSHIPPING` is set to 1.

PayPal review page with a required address



Suppressing the Buyer's Shipping Address

You can suppress the display of the buyer's shipping address on the PayPal pages. You might want to do this in these cases:

- You are selling a product or service that does not require shipping.
- You prefer to handle addresses completely on your own and do not want to let buyers choose from their PayPal address book.

To suppress the display of the buyer's shipping address, set the `NOSHIPPING` parameter to 1 in the call to `SetExpressCheckout`. No shipping address displays on the PayPal pages whatsoever.

The following example suppresses the shipping address.

Request Parameters

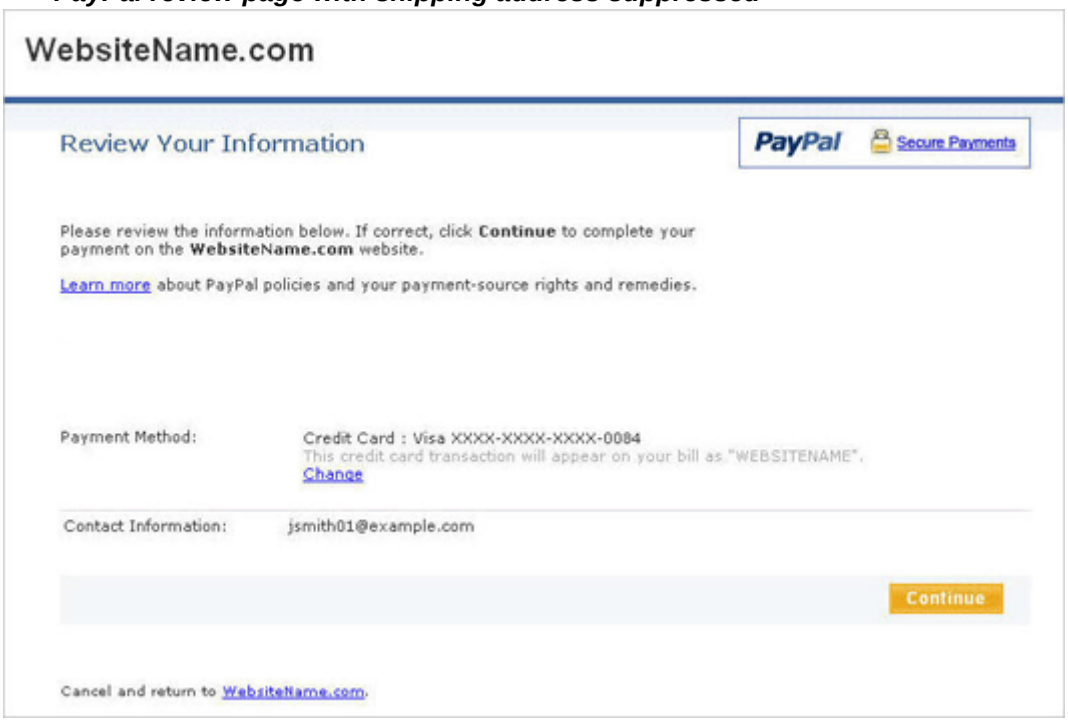
[requiredSecurityParameters]
&**METHOD=SetExpressCheckout**
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&CURRENCYCODE=USD
&PAYMENTACTION=Sale
&**NOSHIPPING=1**
&SHIPTOSTREET=1 Main St
SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&SHIPTOPHONENUM=408-967-4444

Response Parameters

[successResponseFields]
&TOKEN=EC-17C76533PL706494P

The figure below shows the PayPal review page when NOSHIPPING is set to 1.

PayPal review page with shipping address suppressed


WebsiteName.com
Review Your Information
PayPal
Secure Payments
Please review the information below. If correct, click Continue to complete your payment on the WebsiteName.com website.
Learn more about PayPal policies and your payment-source rights and remedies.
Payment Method:
Credit Card : Visa XXXX-XXXX-XXXX-0084
This credit card transaction will appear on your bill as "WEBSITENAME".
Change
Contact Information:
jsmith01@example.com
Continue
Cancel and return to WebsiteName.com.

Shipping Address Override

You can override the buyer's shipping address stored on PayPal. You would want to do this if, for example, your website registration already requested the customer shipping address.

Overriding the shipping address stored on PayPal replaces it with one you specify in the call to `SetExpressCheckout`. The buyer cannot edit the overridden address.

NOTE: If you prefer to override addresses, it is recommended that you do not set confirmed addresses as required, as described in Confirmed Address.

To override the shipping address:

1. Set the `ADDROVERRIDE` parameter to 1 in the call to `SetExpressCheckout`.
2. Set the following shipping address parameters in the call to `SetExpressCheckout` to the address values you want to use for the new address.
   - `SHIPTONAME`
   - `SHIPTOSTREET`
   - `SHIPTOCITY`
   - `SHIPTOSTATE (Optional)`
   - `SHIPTOCOUNTRYCODE`
   - `SHIPTOZIP`
   - `SHIPTOSTREET2` (Optional)

The following example overrides the shipping address with the address values shown.

Request Parameters

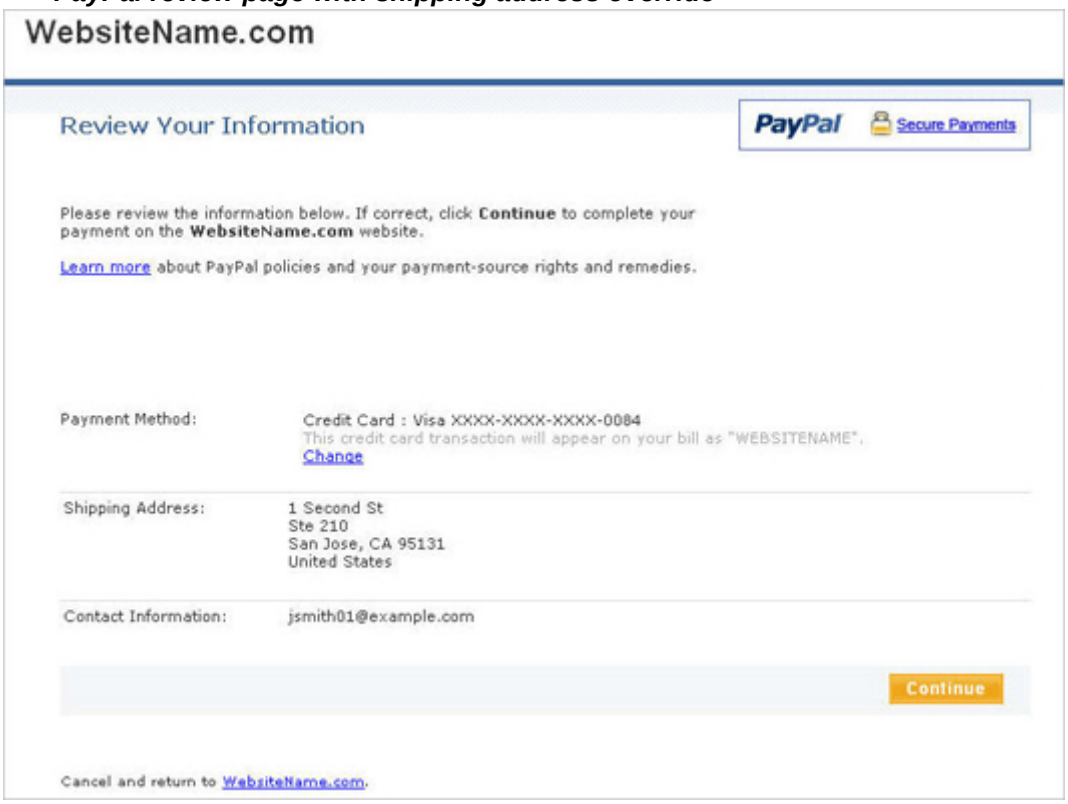
[requiredSecurityParameters]
&**METHOD=SetExpressCheckout**
&RETURNURL=https://...
&CANCELURL=https://...
AMT=10.00
&CURRENCYCODE=USD
&PAYMENTACTION=Sale
&ADDROVERRIDE=1
&**SHIPTOSTREET=1 Second St**
&**SHIPTOSTREET2=Ste 210**
&**SHIPTOCITY=San Jose**
&**SHIPTOSTATE=CA**
&**SHIPTOCOUNTRYCODE=US**
&**SHIPTOZIP=95131**
&SHIPTOPHONENUM=408-967-4444

Response Parameters

[successResponseFields]
&TOKEN=EC-57K68322WE343022B

The figure below shows the PayPal Review page when the shipping address parameters are overridden by the values shown in the above example.

PayPal review page with shipping address override


WebsiteName.com
Review Your Information
PayPal
Secure Payments
Please review the information below. If correct, click Continue to complete your payment on the WebsiteName.com website.
Learn more about PayPal policies and your payment-source rights and remedies.
Payment Method:
Credit Card : Visa XXXX-XXXX-XXXX-0084
This credit card transaction will appear on your bill as "WEBSITENAME".
Change
Shipping Address:
1 Second St
Ste 210
San Jose, CA 95131
United States
Contact Information:
jsmith01@example.com
Continue
Cancel and return to WebsiteName.com.

Automatically Filling Out the PayPal Login Page

When you pass the buyer's shipping address and contact information (telephone number and email address) parameters in the call to `SetExpressCheckout`, PayPal automatically fills out the account creation form fields for the buyer on the PayPal Login page.

After the call to `SetExpressCheckout`, the buyer is redirected to the PayPal Login page. Buyers having a PayPal account can log in with their email address and password. Buyers who do not have an account can create one by filling out the form on this page.

To facilitate filling out the form, you can have PayPal automatically fill out the billing address and contact information by passing the shipping address, telephone number, and email address in the call to `SetExpressCheckout`, as shown in the example below:

Request Parameters

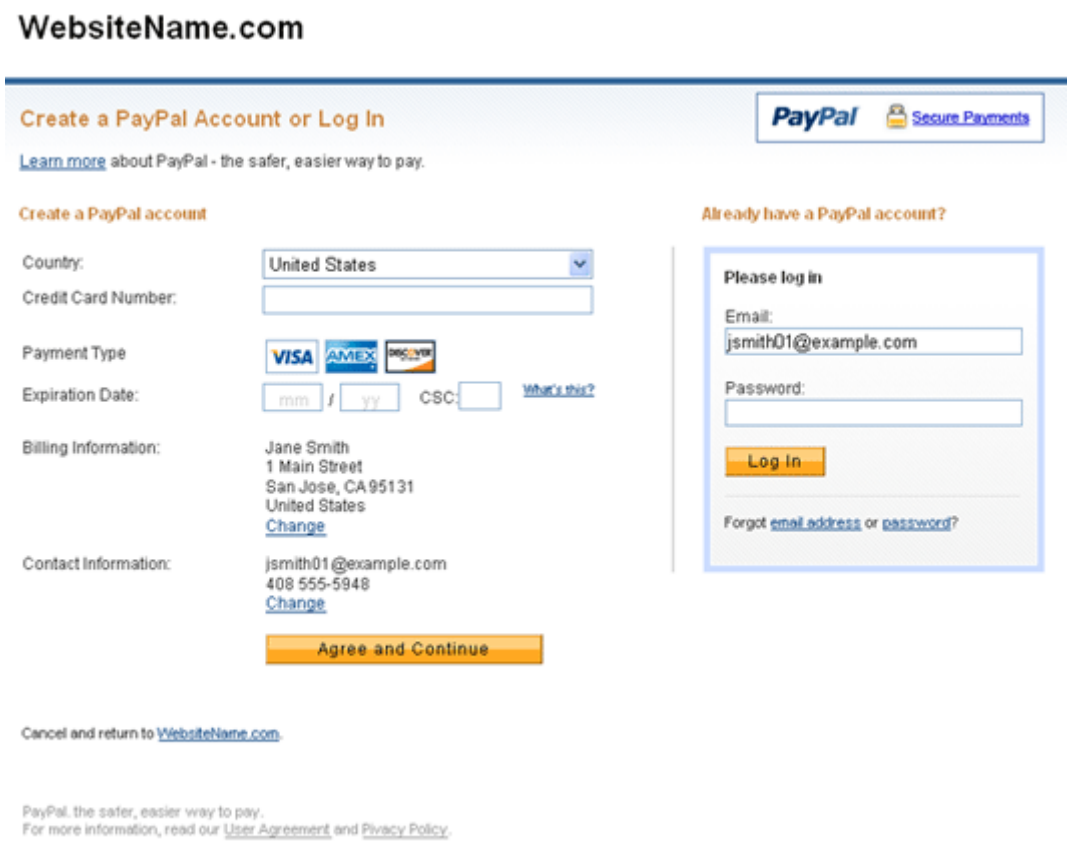
[requiredSecurityParameters]
&**METHOD=SetExpressCheckout**
&RETURNURL=https://...
&CANCELURL=https://...
&AMT=10.00
&PAYMENTACTION=Sale
&**SHIPTOSTREET=1 Main Street**
&**SHIPTOCITY=San Jose**
&**SHIPTOSTATE=CA**
&**SHIPTOCOUNTRYCODE=US**
&**SHIPTOZIP=95131**
&**EMAIL=jsmith01@example.com**
&**SHIPTOPHONENUM=408-559-5948**

Response Parameters

[successResponseFields]
&TOKEN=EC-6UA07551EA393551U

As a convenience, PayPal fills out the billing address fields with the buyer's shipping address as shown in the example figure below. The buyer can edit the information by clicking the Change links.

Pre-populated PayPal Login page



User Confirms Order on PayPal

If you do not require the buyer to explicitly review and confirm the order on your site, you can configure Express Checkout such that the user commits the payment on PayPal.This reduces a step in the checkout flow.

You may want to eliminate your “Review Order” page if there is no additional information you want to collect from the user before they complete the transaction. It is recommended that you collect the information *after* the buyer completes the purchase.

You should evaluate each Express Checkout flow separately through the cart page and through the payment methods page. In most checkout flow implementations, the payment methods page is the last page the buyer sees before commiting to a transaction. If this true in your implementation, you can use this feature to streamline the buyer experience. The feature informs the buyer that they are commiting to the transaction if they proceed.

To invoke this feature, use the `useraction` variable in the `SetExpressCheckout` call. Setting `useraction` to `commit` sets the button text on the PayPal Payment Review page to

read **Pay Now**. (After the buyer returns from the PayPal site, you must call `DoExpressCheckoutPayment` to actually complete the transaction.)

To display **Pay Now** on the button:

1. Get the token from the response to `SetExpressCheckout`.

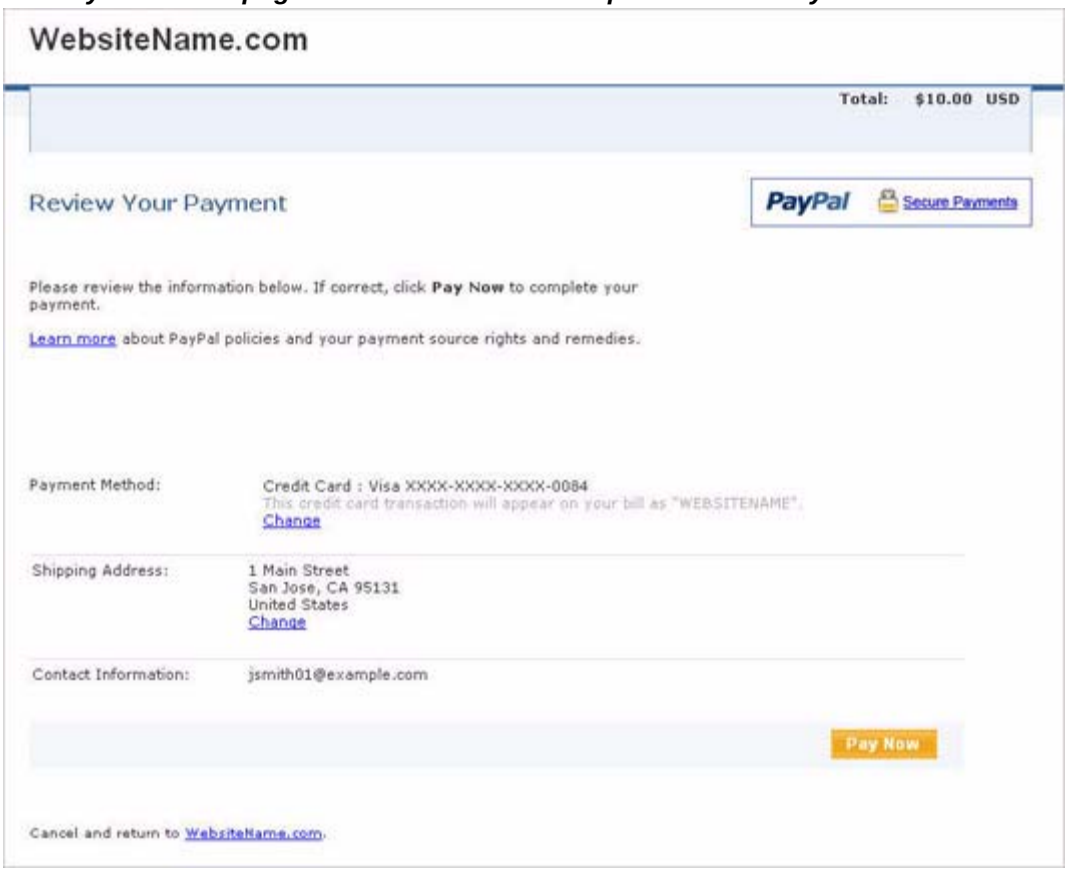
   The response to `SetExpressCheckout` is the buyer's token. For example, if the value of `RETURNURL` passed to `SetExpressCheckout` is `https://www.websitename.com/snagECvalues`, the URL to which PayPal redirects appears as `https://www.websitename.com/snagECvalues?token=EC-0W8920957N684880R`

2. Add the token and the desired *useraction* value (`continue` or `commit`) as name-value pairs to the following URL:

   https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&**token**=*valueFromSetExpressCheckoutResponse*&**useraction=commit**

The figure below shows the PayPal Review page when `useraction=commit`.

PayPal review page when user commits to purchase on PayPal



7 Implementing the Instant Update API

The Instant Update API is a callback you can use to obtain the buyer's shipping address.

- About the Instant Update API
- How the Callback Works in the Express Checkout Flow
- Following Instant Update API Best Practices
- Setting Up the Callback
- Using the Callback

About the Instant Update API

The Instant Update API is a server call to your web service that provides an instant update to the PayPal review page. It enables you to specify a URL for PayPal to call your web service with the buyer's shipping address, so you can provide the buyer with more detailed shipping, insurance, and tax information.

NOTE: The Instant Update API enhances the Express Checkout flow from the shopping cart page.

The Instant Update API works as follows:

1. As the buyer logs in to PayPal, the PayPal server calls a URL to your web service with the buyer's default shipping address as stored in the PayPal system.
2. Your callback server responds with the available shipping options for that address, along with any insurance options and tax adjustments on the order.
3. PayPal displays this information on the Review page so buyers can choose from the options.
4. The buyer's final choices are returned in the `GetExpressCheckoutDetails` response.

Integration Steps

Integrating the Instant Update API requires some preparation and modification to the Express Checkout API calls.

To integrate the server API, follow these steps:

1. Set up a secure, fast web service to accept HTTP requests from PayPal. On the live site, it needs to be secured by means of SSL.
2. Enable the callback service to process PayPal requests and send back responses.

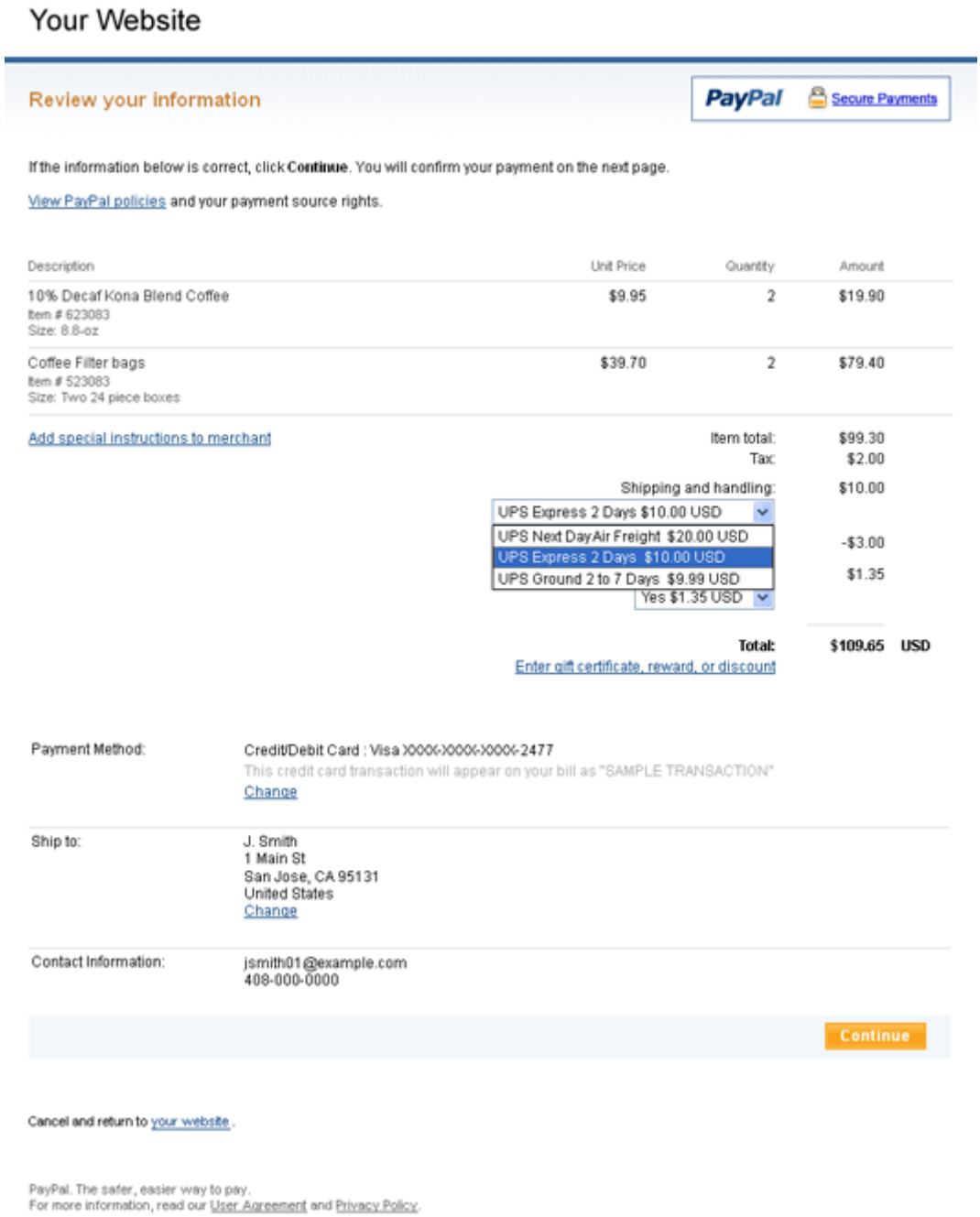
3. Modify the existing Express Checkout API calls to accommodate new parameters.
   - Send the callback URL, shipping, insurance, and tax information to PayPal in the call to `SetExpressCheckout`.
   - Call `GetExpressCheckoutDetails` to obtain the buyer's final choices for shipping and insurance, if applicable.
   - Call `DoExpressCheckoutPayment` with the buyer's final selections.
4. Eliminate your shipping options page.
5. Test your integration for the callback and flat rate shipping options.

Post-Integration Checkout Experience

After you have integrated the Instant Update API, you can display the shipping options, insurance options, and tax on the PayPal Review page.

The shipping and insurance options appear in drop-down menus as shown below.

PayPal Review page with shipping options, insurance, and tax

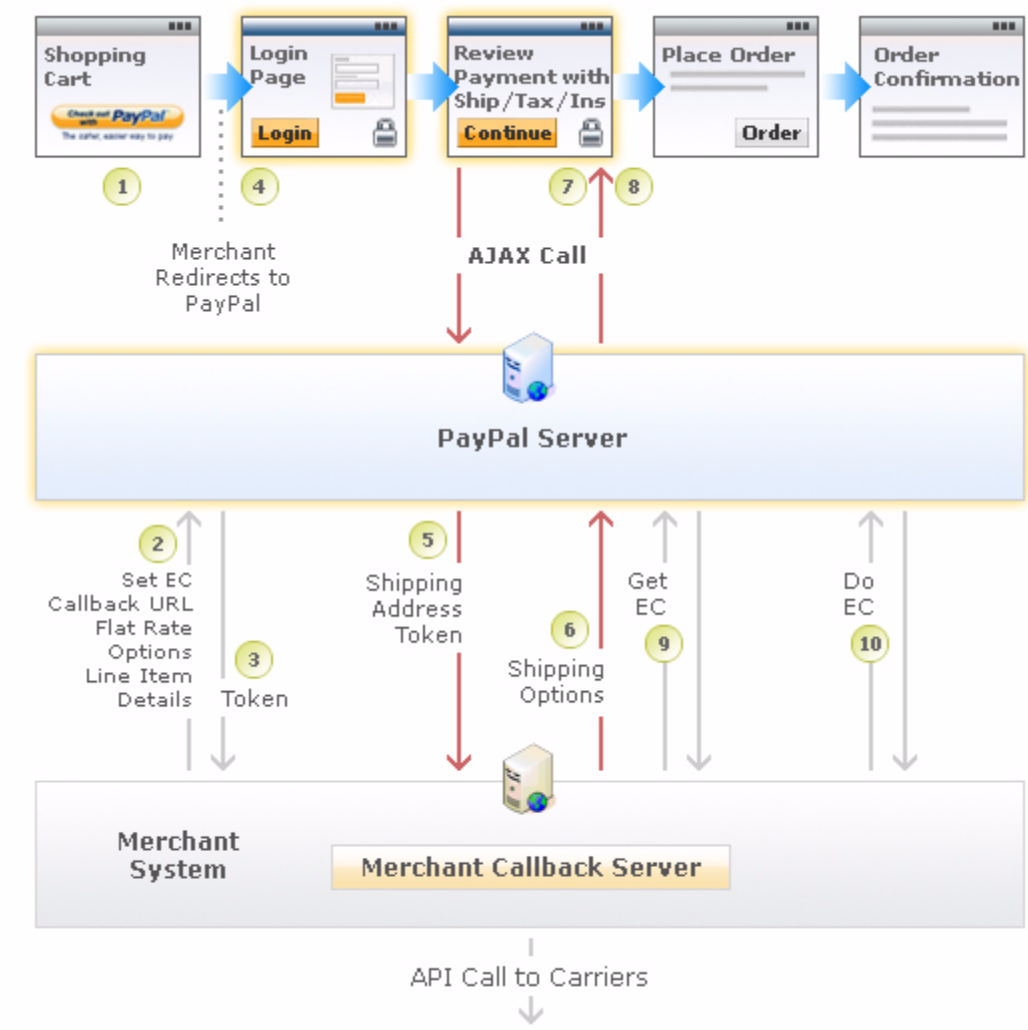


You can control what options are presented to the buyer as the options can be instantly updated on the page.

How the Callback Works in the Express Checkout Flow

The figure below shows how the callback integrates into the Express Checkout flow.

Callback integrated into Express Checkout flow



From left to right, the following events are represented. Text in boldface describes events supporting the callback.

1. The Express Checkout flow is initiated on your shopping cart page when the buyer clicks the Checkout with PayPal button.
2. In the call to the `SetExpressCheckout` API operation, you provide the URL where PayPal can call your callback server**, the flat rate shipping options, and cart line-item details**.
3. You receive a token in the response.

4. The buyer is redirected to PayPal.
5. When the buyer first logs in to the PayPal site, PayPal obtains the buyer's shipping address and sends it in the callback request (red down arrow) to your callback server at the specified URL.

   NOTE: If the buyer changes their shipping address on the PayPal Review page, PayPal will make subsequent calls to the callback request.

6. **You respond to the callback (red up arrow) with the shipping option rates based on the buyer's shipping address. You can also adjust the tax amount and send insurance options. Depending on your business processes, you may send an API call to your carrier to calculate the rates and options based on the shipping address.**
7. **PayPal updates the Review Page to show the options and rates based on your response.**
8. The buyer makes final selections and clicks **Continue**.
9. You must call `GetExpressCheckoutDetails` to obtain the buyer's final shipping option selections.
10. You call `DoExpressCheckoutPayment` to perform the transaction.

Following Instant Update API Best Practices

When implementing the Instant Update API, PayPal recommends that you follow best practices.

Use the following best practices as a checklist as you integrate the API:

- **Meet the Pre-requisites:** Provide order line-item details to take advantage of the Instant Update API. The example in this document uses the order line-item details example described in "PayPal Review Page Order Details" on page 45.
- **Streamline the Checkout Flow:** Existing partners and merchants with Express Checkout integrations can eliminate the current shipping options page.
- **Use the Default Callback Timeout:** Use the default callback response time of 3 seconds.
- **Follow PayPal-Defined Semantics and Syntax:** Adhere to well-formed variable names and syntax rules in the callback response to PayPal. If any errors occur in the response, PayPal will use the flat-rate shipping options.
- **Call GetExpressCheckoutDetails:** You must call `GetExpressCheckoutDetails` to find out what options the buyer selected on the PayPal Review page.
- **Ensure a Consistent and Good Buyer Experience:** When the flat-rate shipping options are used, you should honor the rates to ensure a consistent and good buyer experience.
- **Localize Shipping Options:** Return the localized shipping option labels based on the buyer's country and locale that PayPal sends in the callback request.

Setting Up the Callback

To set up the callback, you establish a connection with PayPal by providing the location where PayPal can call your web service, along with your shipping options.

To start, you must build and operate a secure, reliable, and fast web service that computes the shipping options based on the business rules. To verify that the callback request is indeed originating from PayPal, you can check to see if the request originates from *.paypal.com as described in the go-live checklist at this location.

In the sandbox environment, you can use both HTTP and HTTPS. PayPal will validate the following information about the callback:

- The callback URL must start with HTTPS for production integration.
- The callback URL must start with HTTP or HTTPS for sandbox integration.

In the call to `SetExpressCheckout`, you must complete the steps 1 through 3 below. Steps 4 and 5 are optional:

1. Provide line-item details for the merchandise the buyer selected. For details on how to integrate line-item details, see Chapter 7, "Customizing Express Checkout," in the Express Checkout Integration Guide.
2. Provide the URL to your web service for the callback. PayPal validates the URL as described above.
3. Provide values for the flat rate shipping options. For each option, specify
   - Option name (`L_SHIPPINGOPTIONNAME`*n*)
   - Option label (`L_SHIPPINGOPTIONLABEL`*n*)
   - Option amount (`L_SHIPPINGOPTIONAMOUNT`*n*)
   - The shipping option to appear in the drop-down menu as the default (`true`) (`L_SHIPPINGOPTIONISDEFAULT`*n*).

     NOTE: Set `L_SHIPPINGOPTIONISDEFAULT`*n* to `true` (default) for one and only one shipping option. Set `L_SHIPPINGOPTIONISDEFAULT`*n* to false for each of the remaining options.
   - If required, an adjusted value for `TAXAMT`
   - If required, an adjusted value `INSURANCEAMT`
4. If necessary to adjust the callback timeout (default: 3 seconds), provide a value from 1 to 6 for the `CALLBACKTIMEOUT` parameter.
5. Optionally, provide values for any of the shipping option description details fields listed below:
   - Option weight (`L_ITEMWEIGHTVALUE`*n*, `L_ITEMWEITHTUNIT`*n*)
   - Option height (`L_ITEMHEIGHTVALUE`*n*, `L_ITEMHEIGHTUNIT`*n*)

- Option length (`L_ITEMLENGTHVALUE`*n*, `L_ITEMLENGTHUNIT`*n*)
- Option width (`L_ITEMWIDTHVALUE`*n*, `L_ITEMWIDTHUNIT`*n*)

GetExpressCheckoutDetails and DoExpressCheckoutPayment Changes

When you implement the callback, you need to call `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

`GetExpressCheckoutDetails` and `DoExpressCheckoutPayment` include new parameter fields in support of the Instant Update API.

You must call the `GetExpressCheckoutDetails` API operation to obtain the buyer's final shipping option selections. `GetExpressCheckoutDetails` has been updated to return the buyer's selections.

Because the cart information passed in the call to `SetExpressCheckout` is only relevant for display on the PayPal Review page, you must call the `DoExpressCheckoutPayment` API operation with the updated shipping, insurance, and tax data to ensure the buyer sees it upon redirect to your website.

Other Considerations

When you implement the callback, there are other issues you must take into consideration.

These include:

- Callback response errors
- Maximum shipping options
- Callback times out

Callback Response Errors

If there are any callback response errors, PayPal responds by displaying the flat rate shipping options on the PayPal Review page. To obtain the richer set of options available through the callback, exercise care in the syntax and values you specify and test the callback integration.

Maximum Shipping Options

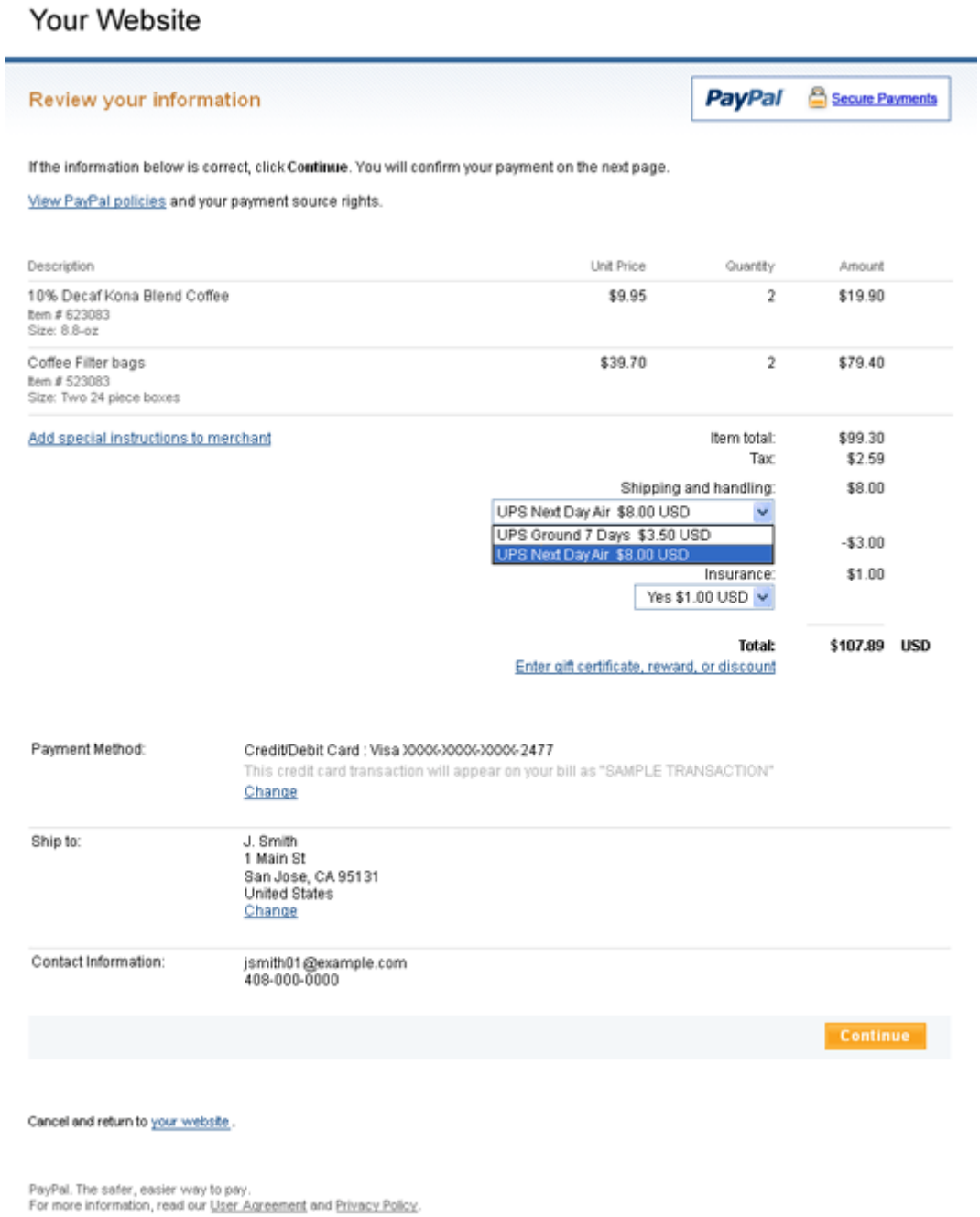
You can specify up to a maximum of 10 shipping options for the flat rate options in the call to `SetExpressCheckout` and for the detailed options based on shipping address in the callback response.

Callback Times Out

If the callback does not return within the timeout period, PayPal displays the flat rate shipping options you specified in the call to `SetExpressCheckout` in the drop-down menu on the PayPal Review page.

The PayPal Review page in the figure below shows two flat rate shipping options the buyer can choose from if the callback times out. Note also that a flat rate amount of $1.00 is offered for insurance.

PayPal Review page when callback times out



Using the Callback

To use the callback, you add parameter fields to `SetExpressCheckout`, provide PayPal a URL for sending a callback request, and send PayPal the callback response in Name-Value pair (NVP) format.

SetExpressCheckout

In the call to `SetExpressCheckout`, set the following parameters:

- Set the `CALLBACKURL` field to the URL where PayPal can call your web service. PayPal makes the HTTPS callback request each time either of the following events occur:
  - The buyer changes their shipping address
  - The buyer enters a new shipping address
- Provide values for the following required parameters:
  - Provide values for the line-item details parameters such as `L_NAME`*n*, `L_NUMBER`*n*, `L_DESC`*n*, `L_AMT`*n*, and `L_QTY`*n* shown in the example below.
  - Provide values for the flat-rate shipping options: *n*, `L_SHIPPINGOPTIONISDEFAULT`*n*, `L_SHIPPINGOPTIONNAME`*n*, `L_SHIPPINGOPTIONLABEL`*n*, and `L_SHIPPINGOPTIONAMOUNT`*n*.
  - Set `SHIPPINGAMT` to the amount set for the default flat-rate shipping option.

    If, for example, `L_SHIPPINGISDEFAULT1=true` and `L_SHIPPINGOPTIONAMOUNT1=8.00`, then `SHIPPINGAMT=8.00`
  - Set `MAXAMT` to the expected maximum total amount of the complete order.

    It is recommended that the maximum total amount be slightly greater than the sum of the line-item order details, tax, and the shipping option of greatest value.
- Optionally provide values for the following parameters:
  - Set `INSURANCEOPTIONOFFERED` to `true` to inform PayPal that you are offering insurance options. Otherwise, set `INSURANCEOPTIONSOFFERED` to `false`.
  - Set line-item description details such as `L_ITEMWEIGHTUNIT1` and `L_ITEMWEIGHTVALUE1` shown in the example below.
  - Set `CALLBACKTIMEOUT` to the amount of time in seconds to process the callback. By default, `CALLBACKTIMEOUT` is `3`. You can specify a value in the range of `1` to `6` inclusive.

This is an example `SetExpressCheckout` request. The above parameters appear in boldface text.

Request Parameters:

[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTACTION=Sale
&NAME=J Smith
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOZIP=95131
&L_NAME0=10% Decaf Kona Blend Coffee
&L_NUMBER0=623083
&L_DESC0=Size: 8.8-oz
&L_AMT0=9.95
&L_QTY0=2
&L_NAME1=Coffee Filter bags
&L_NUMBER1=6230
&L_DESC1=Size: Two 24-piece boxes
&L_AMT1=39.70
&L_QTY1=2
&L_ITEMWEIGHTVALUE1=0.5
&L_ITEMWEIGHTUNIT1=lbs
&ITEMAMT=99.30
&TAXAMT=2.59
&MAXAMT=150.00
&SHIPPINGAMT=8.00
&SHIPDISCAMT=-3.00
&AMT=107.89
&CURRENCYCODE=USD
&ALLOWNOTE=1
&CALLBACK=https://...
&CALLBACKTIMEOUT=4
&INSURANCEOPTIONOFFERED=true
&INSURANCEAMT=1.00
&L_SHIPPINGOPTIONISDEFAULT0=false
&L_SHIPPINGOPTIONNAME0=Ground
&L_SHIPPINGOPTIONLABEL0=UPS Ground 7 Days
&L_SHIPPINGOPTIONAMOUNT0=3.50
&L_SHIPPINGOPTIONISDEFAULT1=true
&L_SHIPPINGOPTIONNAME1=UPS Air
&L_SHIPPINGOPTIONlABEL1=UPS Next Day Air Freight
&L_SHIPPINGOPTIONAMOUNT1=8.00

Response Parameters

[successResponseFields]
&TOKEN=EC-17C76533PL706494P

Callback Request

The PayPal sends the parameters in the callback request to the location you specified for `CALLBACKURL`. The callback request parameters include:

- The line-item details you sent in the call to `SetExpressCheckout`. PayPal also sends back any line-item description details you may have specified such as the `L_ITEMWEIGHTUNIT1` and `L_ITEMWEIGHTVALUE1` values shown in the example below. By passing this data back to you, PayPal expedites your callback response by eliminating the need for you to perform a database query to get this information.
- The shipping address of the buyer.

Using the information in the callback request, you can calculate the rates and options yourself or send the information in an API call to your carrier to perform the calculations for you.

This is an example callback request.

```
METHOD=CallbackRequest
&CALLBACKVERSION=57.0
&TOKEN=EC-0EE85728D547104V
&CURRENCYCODE=USD
&LOCALECODE=en_US
&L_NAME0=10% Decaf Kona Blend Coffee
&L_NUMBER0=623083
&L_DESC0=Size: 8.8-oz
&L_AMT0=9.95
&L_QTY0=2
&L_NAME1=Coffee Filter bags
&L_NUMBER1=6230
&L_DESC1=Size: Two 24-piece boxes
&L_AMT1=39.70
&L_QTY1=2
&L_ITEMWEIGHTUNIT1=lbs
&L_ITEMWEIGHTVALUE1=0.5
&SHIPTOSTREET=1 Main St
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRY=US
&SHIPTOZIP=95131
&SHIPTOSTREET2
```

Callback Response

Each time your web service receives a request from PayPal, it must process the request and respond with the appropriate details.

This is an example callback response.

```
METHOD=CallbackResponse
&OFFERINSURANCEOPTION=true
&L_SHIPPINGOPTIONNAME0=UPS Air
&L_SHIPPINGOPTIONLABEL0=UPS Next Day Air Freight
&L_SHIPPINGOPTIONAMOUNT0=20.00
&L_TAXAMT0=2.20
&L_INSURANCEAMOUNT0=1.51
&L_SHIPPINGOPTIONISDEFAULT0=false
&L_SHIPPINGOPTIONNAME1=UPS Expedited
&L_SHIPPINGOPTIONLABEL1=UPS Express 2 Days
&L_SHIPPINGOPTIONAMOUNT1=10.00
&L_TAXAMT1=2.00
&L_INSURANCEAMOUNT1=1.35
&L_SHIPPINGOPTIONISDEFAULT1=true
&L_SHIPPINGOPTIONNAME2=UPS Ground
&L_SHIPPINGOPTIONLABEL2=UPS Ground 2 to 7 Days
&L_SHIPPINGOPTIONAMOUNT2=9.99
&L_TAXAMT2=1.99
&L_INSURANCEAMOUNT2=1.28
&L_SHIPPINGOPTIONISDEFAULT2=false
```

8 Immediate Payment

Immediate Payment ensures buyer payment for an item you are selling on eBay at the time the buyer commits to the item, to avoid unpaid items that drive up the operational cost of your business. You integrate Immediate Payment as additional Express Checkout parameters and use your own off-eBay checkout flow.

- "Overview of Immediate Payment" on page 81
- "Integrating Immediate Payment into the Express Checkout Flow" on page 83

NOTE: Immediate Payment is available with API version 60.0 and later.

Overview of Immediate Payment

Immediate Payment enables you to have the buyer pay for an eBay item at the time the buyer commits to it. This feature adds a few new `SetExpressCheckout` request parameters to ensure the eBay item is available to ship to the buyer when `DoExpressCheckoutPayment` is called. The successful transaction returns the eBay transaction ID as an additional parameter value in the `DoExpressCheckoutPayment` response.

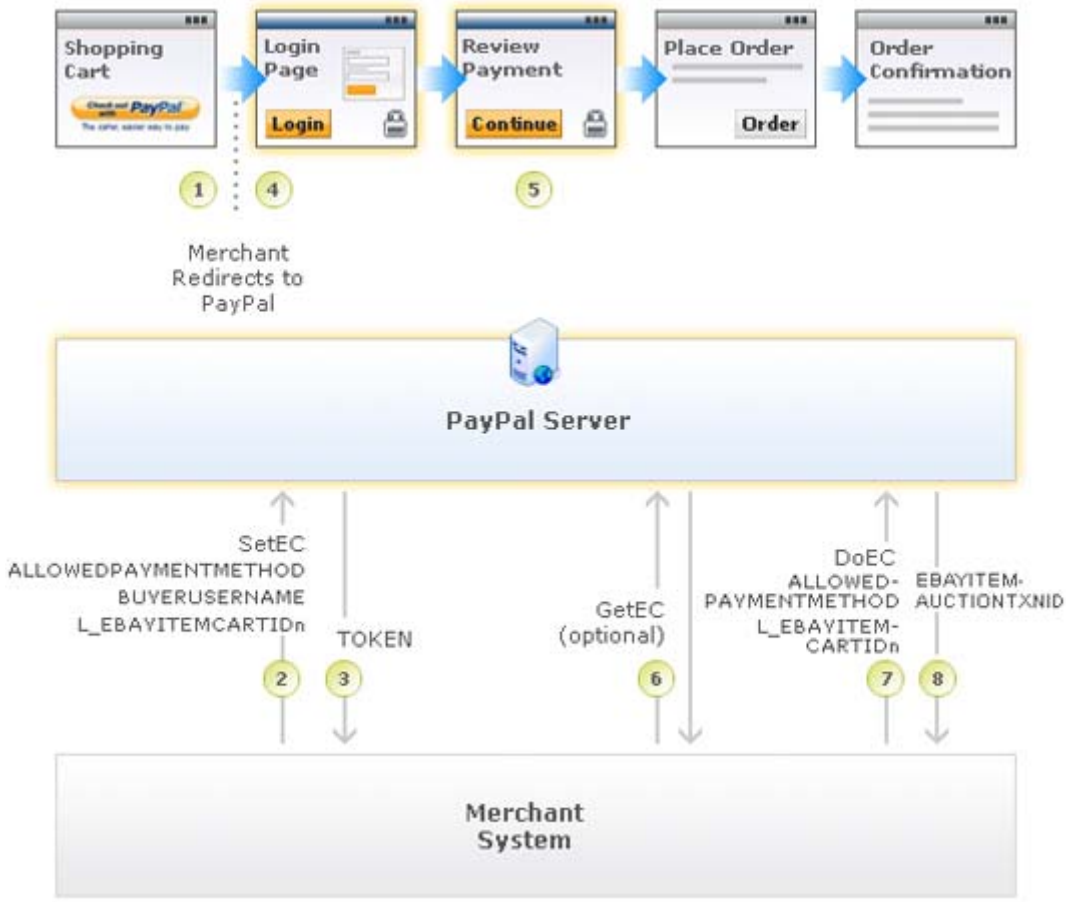
NOTE: Sale is the only payment action supported by Immediate Payment.

Immediate Payment supports instant funding sources only so you can receive payment at the time your buyer commits to the eBay item.

NOTE: PayPal recommends that you do not mix eBay and non-eBay items in an Immediate Payment Transaction.

How Immediate Payment Works in the Express Checkout Flow

After the buyer selects the item on eBay, the buyer is redirected to your website Express Checkout flow. The figure below shows how Immediate Payment integrates into your checkout flow.



From left to right, the following events are represented. The numbered callouts in the figure directly correspond to the numbered comments below:

1. The eBay flow for third-party checkout redirects the buyer from an eBay payment review page to your shopping cart page. On your shopping cart page, the buyer initiates Express Checkout by clicking the **Checkout with PayPal** button.
2. In the call to the `SetExpressCheckout` API operation, you must pass the following Immediate Payment information:
   - `ALLOWEDPAYMENTMETHOD`: This is the payment method type. For immediate payment, the value is `InstantPaymentOnly`.
   - `BUYERUSERNAME`: EBay provides you with this value.
   - `L_EBAYITEMCARTIDn`: EBay provides you with this value.

   Additionally, in the call to the `SetExpressCheckout` API operation, you must pass the following Express Checkout API parameter data:
   - `CHANNELTYPE`: The value of the channel type, which is `eBayItem` .
   - `L_EBAYITEMNUMBERn`: This is the eBay item number.

– Buyer's shipping address.

3. The `SetExpressCheckout` response returns a `TOKEN`.
4. The buyer is redirected to PayPal.
5. The buyer reviews their payment and clicks **Continue**.
6. Optionally, you may call `GetExpressCheckoutDetails`. The `GetExpressCheckoutDetails` response returns information about the buyer.
7. In the call to `DoExpressCheckoutPayment`, you must pass the .
   – `ALLOWEDPAYMENTMETHOD`: This is the payment method type. For immediate payment, the value is `InstantPaymentOnly`.
   – `L_EBAYITEMCARTIDn`: EBay provides you with this value.

   At this time, PayPal checks the availability of the eBay item.
8. If the item is available and payment is successful, the `DoExpressCheckoutPayment` response returns the eBay transaction ID (`EBAYITEMAUCTIONTXNID`).

Integrating Immediate Payment into the Express Checkout Flow

To integrate Immediate Payment into your Express Checkout flow, you add a few new fields to `SetExpressCheckout` and `DoExpressCheckoutPayment`.

The Call to SetExpressCheckout

You must have Express Checkout integrated into your payment solution. It is recommended that you sell one eBay item in a transaction or a quantity of the same item (considered a single eBay listing).

To integrate Immediate Payment into the `SetExpressCheckout` call:

1. Set `ALLOWEDPAYMENTMETHODTYPE` to `InstantPaymentOnly`.

   This blocks all pending funding sources and transactions that end up in a pending state.
2. Pass `BUYERUSERNAME` and `L_EBAYITEMCARTIDn`..
3. Set `CHANNELTYPE` to `eBayItem`..
4. Set `L_EBAYITEMNUMBERn` to the number of the eBay item.
5. Specify the buyer's shipping address.
6. Be prepared to handle Immediate Payment errors.

For errors you choose to handle, you need to send the buyer error messages appropriate to the situations generating the errors. Say, for example, you determine that the buyer has a mixed

cart of eBay and non-eBay items.To allow the buyer to check out *just* the non-Immediate Payment items, you must provide an appropriate message to the buyer, remove the Immediate Payment item, and retry the transaction.

The Call to DoExpressCheckoutPayment

To integrate Immediate Payment into the `DoExpressCheckoutPayment` call:

1. Set `ALLOWEDPAYMENTMETHODTYPE` to `InstantPaymentOnly`.

   This blocks all pending funding sources and transactions that end up in a pending state.

2. Pass the `L_EBAYITEMCARTIDn`.

3. Be prepared to handle Immediate Payment errors.

   For errors you choose to handle, you need to send the buyer error messages appropriate to the situations generating the errors. Say, for example, you determine that the buyer has a mixed cart of eBay and non-eBay items. To allow the buyer to check out *just* the non-Immediate Payment items, you must provide an appropriate message to the buyer, remove the Immediate Payment item, and retry the transaction.

If payment is successful, the `DoExpressCheckoutPayment` response returns the eBay transaction ID (`EBAYITEMAUCTIONTXNID`).

9 Handling Payment Settlements

You can use PayPal API operations to handle the capture of payments authorized using Express Checkout and manage Express Checkout authorization and order payment actions.

- Sale Payment Action
- Authorization Payment Action
- Order Payment Action

Sale Payment Action

A *sale* payment action represents a single payment that completes a purchase for a specified amount.

A sale is the default Express Checkout payment action; however, you can also specify the following action in your `SetExpressCheckout` and `DoExpressCheckoutPayment` requests:

```
PAYMENTACTION=Sale
```

A sale is the most straightforward payment action. Choose this payment action if the transaction, including shipping of goods, can be completed immediately. To use this payment action:

- The final amount of the payment must be known when you invoke the `DoExpressCheckoutPayment` API operation
- You should intend to fulfill the order immediately, such as would be the case for digital goods or for items you have in stock for immediate shipment

After you execute the `DoExpressCheckoutPayment` API operation, the payment is complete and further action is unnecessary. You cannot capture a further payment or void any part of the payment when you use this payment action.

Authorization Payment Action

An *authorization* payment action represents an agreement to pay and places the buyer's funds on hold for up to three days.

To set up an authorization, specify the following payment action in your `SetExpressCheckout` and `DoExpressCheckoutPayment` requests:

```
PAYMENTACTION=Authorization
```

An authorization enables you to capture multiple payments up to 115% of, or USD $75 more than, the amount you specify in the `DoExpressCheckoutPayment` request. Choose this payment action if you need to ship the goods before capturing the payment or if there is some reason not to accept the payment immediately.

The *honor period*, for which funds can be held, is three days. The *valid period*, for which the authorization is valid, is 29 days. You can reauthorize the 3-day honor period at most once within the 29-day valid period.

You can void an authorization, in which case, the uncaptured part of the amount specified in the `DoExpressCheckoutPayment` request becomes void and can no longer be captured. If no part of the payment has been captured, the entire payment becomes void and nothing can be captured.

API operations associated with authorization payment actions

API Operation	Description
`DoCapture`	Capture an authorized payment
`DoReauthorization`	Reauthorize a payment
`DoVoid`	Void an order or an authorization

Order Payment Action

An *order* payment action represents an agreement to pay one or more authorized amounts up to the specified total over a maximum of 29 days.

To set up an order, specify the following payment action in your `SetExpressCheckout` and `DoExpressCheckoutPayment` requests:

```
PAYMENTACTION=Order
```

An order enables you to create multiple authorizations over the 29 days; each authorization you create places the buyer's funds on hold for up to three days. You can capture multiple payments for each authorization, up to 115% of, or USD $75 more than, the amount you specify in the `DoExpressCheckoutPayment` request.

NOTE: The default number of child authorizations in your PayPal account is 1. To do multiple authorizations please contact PayPal to request an increase.

This payment action provides the most flexibility and should be used when either a sale or one authorization plus one reauthorization do not meet your needs. Situations in which orders are appropriate include the handling of

- Back orders, in which available merchandise is sent immediately and the remaining merchandise is sent when available, which may include more than two shipments
- Split orders, in which merchandise is sent in more than one shipment, perhaps to different addresses, and you want to collect a payment for each shipment

- Drop shipments, which are shipments from other vendors for which you accept the payment

You cannot reauthorize an authorization. You handle the need to reauthorize, for example when the hold period or valid period of an authorization expires, simply by creating another authorization.

You can void an order or an authorization created from the order. If you void an order, the uncaptured part of the amount specified in the `DoExpressCheckoutPayment` request becomes void and can no longer be captured. If no part of the payment has been captured, the entire payment becomes void and nothing can be captured.

If you void an authorization associated with the order, the uncaptured part of the amount specified in the authorization becomes void and can no longer be captured. If no part of the authorization has been captured, the entire authorized payment becomes void.

API operations associated with order payment actions

API Operation	Description
`DoAuthorization`	Authorize a payment
`DoCapture`	Capture an authorized payment
`DoVoid`	Void an order or an authorization

10 Handling Recurring Payments

Set up a recurring payment to handle subscription and other payments that occur on a fixed schedule.

- Recurring Payments Demo
- How Recurring Payments Work
- Recurring Payments Terms
- Recurring Payments With Express Checkout
- Options for Creating a Recurring Payments Profile
- Recurring Payments Profile Status
- Getting Recurring Payments Profile Information
- Modifying a Recurring Payments Profile
- Billing the Outstanding Amount of a Profile
- Recurring Payments Notifications

How Recurring Payments Work

When you create recurring payments for a buyer, you create a *recurring payments profile*. The profile contains information about the recurring payments, including details for an optional trial period and a regular payment period. Each of these subscription periods contains information about the payment frequency and payment amounts, including shipping and tax, if applicable.

After a profile is created, PayPal automatically queues payments based on the billing start date, billing frequency, and billing amount, until the profile expires or is canceled by the merchant. The buyer can also cancel the recurring payments profile.

Note that the queued payments are funded using the normal funding source hierarchy within the buyer's PayPal account.

After the recurring payments profile is created, you can view recurring payments details or cancel the recurring payments profile from your PayPal account.You can also access recurring payments reports from the PayPal **Business Overview** page.

Also, after creating a recurring payments profile, you can use the Recurring Payments API to do the following:

- Get information details about a recurring payments profile.
- Change the status of a recurring payment profile.
- Update the details of the recurring payments profile.

- Bill the outstanding amount of the recurring payments profile.

Limitations

The current release of the Recurring Payments API has the following limitations:

- A profile can have at most one optional trial period and a single regular payment period.
- The profile start date may not be earlier than the profile creation date.

Recurring payments with Express Checkout also has the following limitations:

- At most ten recurring payments profiles can be created during checkout.
- You can only increase the profile amount by 20% in each 180-day interval after the profile is created.

Recurring Payments Terms

Some terms are commonly used by PayPal in the context of recurring payments.

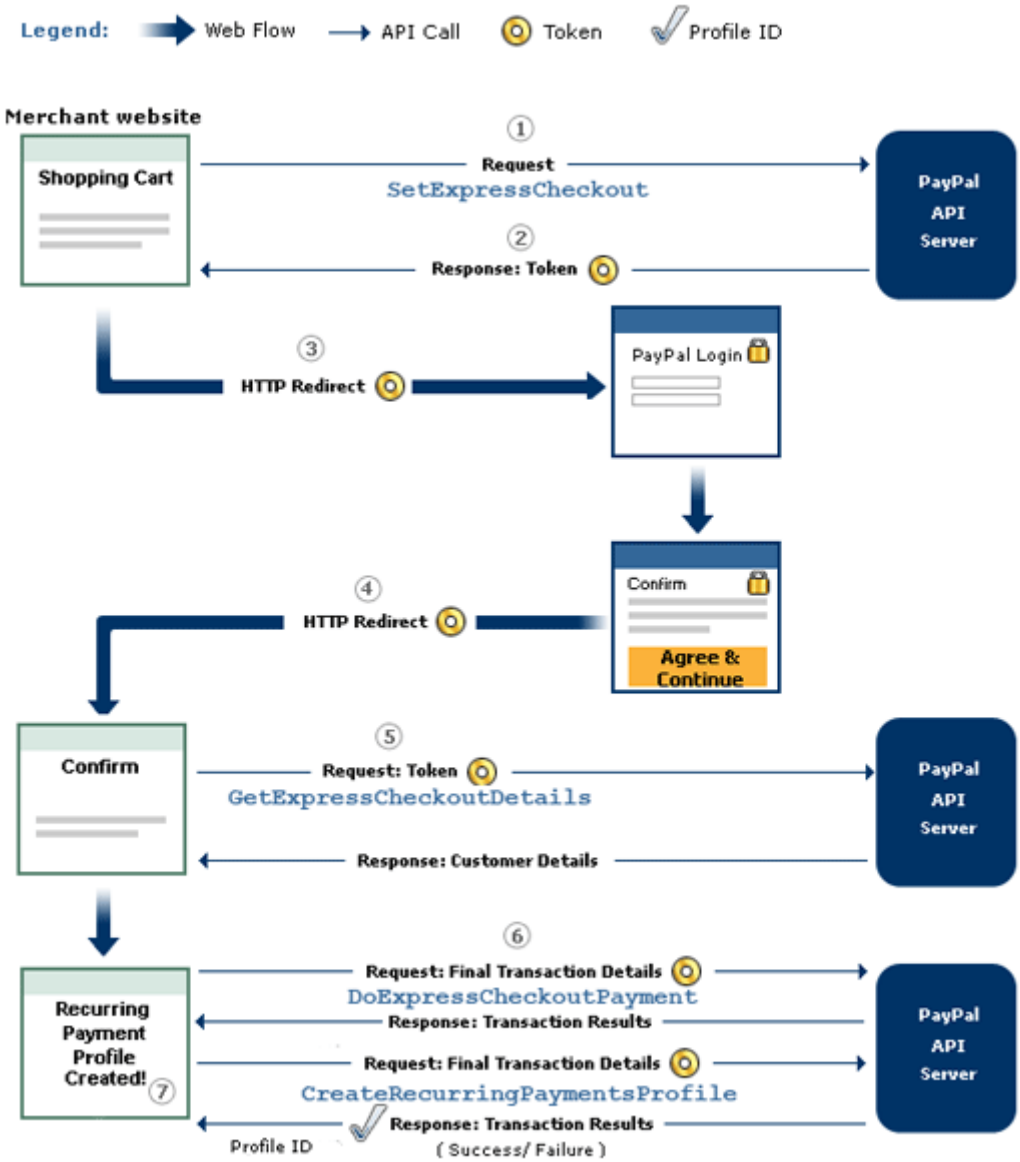
Recurring payments terms

Term	Definition
Recurring payments profile	Your record of a recurring transaction for a single customer. The profile includes all information required to automatically bill the buyer a fixed amount of money at a fixed interval.
Billing cycle	One payment is made per billing cycle. Each billing cycle is made up of two components. • The billing period specifies the unit to be used to calculate the billing cycle (such as days or months). • The billing frequency specifies the number of billing periods that make up the billing cycle. For example, if the billing period is Month and the billing frequency is 2, the billing cycle will be two months. If the billing period is Week and the billing frequency is 6, the payments will be scheduled every 6 weeks.
Regular payment period	The main subscription period for this profile, which defines a payment amount for each billing cycle. The regular payment period begins after the trial period, if a trial period is specified for the profile.
Trial period	An optional subscription period before the regular payment period begins. A trial period may not have the same billing cycles and payment amounts as the regular payment period.
Payment amount	The amount to be paid by the buyer for each billing cycle.
Outstanding balance	If a payment fails for any reason, that amount is added to the profile's outstanding balance.
Profile ID	An alphanumeric string (generated by PayPal) that uniquely identifies a recurring profile.

Recurring Payments With Express Checkout

During the Express Checkout flow, you can create one or more recurring payments and mix recurring payments with other purchases.

The following diagram illustrates the typical processing flow to create recurring payments during checkout.



The circled numbers in the diagram correspond to the following steps:

Recurring payments processing flow

Step	Merchant...	PayPal...
1	Calls `SetExpressCheckout` with one or more billing agreement details in the request	
2		Returns a token, which identifies the transaction, to the merchant.
3	Redirects buyer's browser to: `https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&token=`*<token returned by SetExpressCheckout>*	
		Displays login page. Allows user to select payment options and shipping address.
4		Redirects buyer's browser to `returnURL` passed to `SetExpressCheckout` if buyer agrees to payment description.
5	Calls `GetExpressCheckoutDetails` to get buyer information (optional).	
		Returns `GetExpressCheckoutDetails` response.
	Displays merchant review page for buyer.	
6	Calls `DoExpressCheckoutPayment` if the order includes one-time purchases as well as a recurring payment. Otherwise, skip this step.	
		Returns `DoExpressCheckoutPayment` response
	Calls `CreateRecurringPaymentsProfile` one time for each recurring payment item included in the order.	
		Returns `ProfileID` in `CreateRecurringPaymentsProfile` response for each profile successfully created.
7	Displays successful transaction page.	

Initiating the Processing Flow With SetExpressCheckout

As in the Express Checkout flow, the `SetExpressCheckout` request notifies PayPal that you are:

- Initiating an order that can be either a one-time purchase, up to ten recurring payments, or a mixture of a one-time purchase and recurring payments

- Initiating the processing flow to create one or more billing agreements for recurring payments with no associated one-time purchase or recurring payment

NOTE: You can also initiate the processing flow using `SetCustomerBillingAgreement` for orders that contain only a single recurring payment.

Typically, you set the amount of the payment for an Express Checkout transaction when you call the `SetExpressCheckout` request and confirm the amount in the `DoExpressCheckoutPayment` request. If, however, you set the amount of the payment to 0 in the `SetExpressCheckout` request, you can create a billing agreement without initiating a payment. Regardless of whether a payment is included, you must call `DoExpressCheckoutPayment` to complete the transaction.

NOTE: To create a billing agreement without purchase, use Version 54.0 or higher, of the PayPal API.

To set up one or more billing agreements for recurring payments, modify the `SetExpressCheckout` request as follows:

1. Add an `L_BILLINGTYPE`*n* field for each billing agreement you want to create; *n* is a value in the range of `0` to `9`, inclusive. Set the value of each field to `RecurringPayments`.

   ```
   L_BILLINGTYPE0=RecurringPayments
   ```

2. Add an `L_BILLINGAGREEMENTDESCRIPTION`*n* field to correspond to each `L_BILLINGTYPE`*n* field you pass; *n* is a value in the range of `0` to `9`, inclusive. Set the value of each field to the description of the goods or services associated with that billing agreement, for example:

   ```
   L_BILLINGAGREEMENTDESCRIPTION0=Time Magazine subscription
   ```

3. If there is no associated purchase, set `AMT` to `0`.

   ```
   AMT=0
   ```

NOTE: When you create the recurring payments profile, you must include the same `L_BILLINGTYPE`*n* and `L_BILLINGAGREEMENTDESCRIPTION`*n* values in the `CreateRecurringPaymentsProfile` request.

The `SetExpressCheckout` response provides a token that uniquely identifies the transaction for subsequent redirects and API calls.

Redirecting the Buyer's Browser to PayPal

After you receive a successful response from `SetExpressCheckout`, add the `TOKEN` from the `SetExpressCheckout` response as a name/value pair to the following URL, and redirect your buyer's browser to it:

```
https://www.paypal.com/cgi-bin/webscr?cmd=_express_checkout&
token=<value_from_SetExpressCheckoutResponse>
```

For redirecting the buyer's browser to the PayPal login page, PayPal recommends that you use the HTTPS response 302 "Object Moved" with the URL above as the value of the `Location` header in the HTTPS response. Ensure that you use an SSL-enabled server to prevent browser warnings about a mix of secure and insecure graphics.

Getting Buyer Details Using GetExpressCheckoutDetails

The `GetExpressCheckoutDetails` method returns information about the buyer, including name and email address stored on PayPal. You can optionally call this API after PayPal redirects the buyer's browser to the `ReturnURL` you specified in the `SetExpressCheckout` request.

The `GetExpressCheckoutDetails` request has one required parameter, `TOKEN`, which is the value returned in the `SetExpressCheckout` response.

Creating the Profiles With CreateRecurringPaymentsProfile

After your buyer has agreed to the recurring payments billing agreement on your confirmation page, you must call `CreateRecurringPaymentsProfile` to create the profile. If you are creating multiple recurring payments profiles, you must call `CreateRecurringPaymentsProfile` once for each profile to be created.

If the transaction includes a mixture of a one-time purchase and recurring payments profiles, call `DoExpressCheckoutPayment` to complete the one-time purchase transaction, and then call `CreateRecurringPaymentsProfile` for each recurring payment profile to be created.

IMPORTANT: The recurring payments profile is not created until you receive a success response from the `CreateRecurringPaymentsProfile` call.

The `CreateRecurringPaymentsProfile` response contains a Profile ID, which is an encoded string that uniquely identifies the recurring payments profile.

Options for Creating a Recurring Payments Profile

You can create a recurring payments profile that allows a regular payment period, an optional trial period, an initial payment, and other options.

Specifying the Regular Payment Period

Each recurring payments profile has a regular payment period that defines the amount and frequency of the payment. The following table lists the required fields for specifying the regular payment period.

Required fields for specifying a regular payment period

NVP	SOAP	Description
PROFILESTARTDATE	RecurringPaymentsProfileDetails. BillingStartDate	The date when billing for this profile begins. **NOTE:** The profile may take up to 24 hours for activation.
BILLINGPERIOD	ScheduleDetails. PaymentPeriod. BillingPeriod	The unit of measure for the billing cycle. Must be one of: • Day • Week • SemiMonth • Month • Year
BILLINGFREQUENCY	ScheduleDetails. PaymentPeriod. BillingFrequency	Number of billing periods that make up one billing cycle. **NOTE:** The combination of billing frequency and billing period must be less than or equal to one year. **NOTE:** If the billing period is SemiMonth., the billing frequency must be 1.
AMT	ScheduleDetails. PaymentPeriod. Amount	Amount to bill for each billing cycle.

You can optionally include a value for TOTALBILLINGCYCLES (SOAP field ScheduleDetails.PaymentPeriod.TotalBillingCycles), which specifies the total number of billing cycles in the regular payment period. If no value is specified or if the value is 0, the payments continue until the profile is canceled or suspended. If the value is greater than 0, the regular payment period will continue for the specified number of billing cycles.

You can also specify an optional shipping amount or tax amount for the regular payment period.

Including an Optional Trial Period

You can optionally include a trial period in the profile by specifying the following fields in the CreateRecurringPaymentsProfile request. The following table lists the required fields for creating an optional trial period.

Required fields for specifying a trial period

NVP	SOAP
TRIALBILLINGPERIOD	ScheduleDetails.TrialPeriod.BillingPeriod
TRIALBILLINGFREQUENCY	ScheduleDetails.TrialPeriod.BillingFrequency
TRIALAMT	ScheduleDetails.TrialPeriod.Amount
TRIALTOTALBILLINGCYCLES	ScheduleDetails.TrialPeriod.TotalBillingCycles

Specifying an Initial Payment

You can optionally specify an initial non-recurring payment when the recurring payments profile is created by including the following fields in the `CreateRecurringPaymentsProfile` request:

Required fields for specifying an initial payment

NVP	SOAP
INITAMT	ScheduleDetails.ActivationDetails.InitialAmount
FAILEDINITAMTACTION	ScheduleDetails.ActivationDetails.FailedInitAmountAction

By default, PayPal will not activate the profile if the initial payment amount fails. You can override this default behavior by setting the `FAILEDINITAMTACTION` field to `ContinueOnFailure`, which indicates that if the initial payment amount fails, PayPal should add the failed payment amount to the outstanding balance due on this recurring payment profile.

If this field is not set or is set to `CancelOnFailure`, PayPal will create the recurring payment profile, but will place it into a pending status until the initial payment is completed. If the initial payment clears, PayPal will notify you by IPN that the pending profile has been activated. If the payment fails, PayPal will notify you by IPN that the pending profile has been canceled.

The buyer will receive an email stating that the initial payment cleared or that the pending profile has been canceled if the profile was created using Express Checkout.

Maximum Number of Failed Payments

By including the `MAXFAILEDPAYMENTS` field in the `CreateRecurringPaymentsProfile` request, you set the number of failed payments allowed before the profile is automatically suspended. You receive an IPN message when the number of failed payments reaches the maximum number specified.

Billing the Outstanding Amount

If a payment fails due to any reason, the amount that was to be billed (including shipping and tax, if applicable) is added to the profile's outstanding balance. Use the `AUTOBILLOUTAMT` field in the `CreateRecurringPaymentsProfile` request to specify whether or not the outstanding amount should be added to the payment amount for the next billing cycle.

Whether or not you choose to include the outstanding amount with the payment for the next billing cycle, you can also use the `BillOutstandingAmount` API to programmatically collect that amount at any time.

Recurring Payments Profile Status

The recurring payments actions you may take, depend on the status of the profile.

A recurring payments profile can have one of the following status values:

- `ActiveProfile`
- `PendingProfile`
- `ExpiredProfile`
- `SuspendedProfile`
- `CancelledProfile`

If the profile is successfully created, it has an `ActiveProfile` status. However, if a non-recurring initial payment fails and `FAILEDINITAMTACTION` is set to `CancelOnFailure` in the `CreateRecurringPaymentsProfile` request, the profile is created with a status of `PendingProfile` until the initial payment either completes successfully or fails.

A profile has a status of `ExpiredProfile` when both the total billing cycles for both the optional trial period and the regular payment period have been completed.

You can suspend or cancel a profile by using the `ManageRecurringPaymentsProfileStatus` API. You can also reactivate a suspended profile. If the maximum number of failed payments has already been reached, however, you will need to increase the number of failed payments before reactivating the profile.

NOTE: You can also suspend, cancel, or reactive a recurring payments profile through the PayPal website.

For recurring payments profiles created with Express Checkout, the buyer receives an email about the change in status of their recurring payment.

Getting Recurring Payments Profile Information

Use the `GetRecurringPaymentsProfileDetails` API to get information about a profile.

NOTE: You can also get information about recurring payments profiles from the PayPal website.

Along with the information that you specified in the `CreateRecurringPaymentsProfile` request, `GetRecurringPaymentsProfileDetails` also returns the following summary information about the profile:

- Profile status
- Next scheduled billing date
- Number of billing cycles completed in the active subscription period
- Number of billing cycles remaining in the active subscription period
- Current outstanding balance
- Total number of failed billing cycles
- Date of the last successful payment received
- Amount of the last successful payment received

Modifying a Recurring Payments Profile

Use the `UpdateRecurringPaymentsProfile` API to modify a recurring payments profile.

NOTE: You can also modify recurring payments profiles from the PayPal website.

You can only modify the following specific information about an active or suspended profile:

- Subscriber name or address
- Past due or outstanding amount
- Whether to bill the outstanding amount with the next billing cycle
- Maximum number of failed payments allowed
- Profile description and reference
- Number of additional billing cycles
- Billing amount, tax amount, or shipping amount

NOTE: You cannot modify the billing frequency or billing period of a profile. You can modify the number of billing cycles in the profile.

NOTE: For recurring payments with Express Checkout, certain updates, such as billing amount, are not allowed within 3 days of the scheduled billing date, and an error is returned.

You can modify the following profile information during the trial period or regular payment period.

- Billing amount (excluding tax and shipping)
- Number of billing cycles

The profile changes take effect with the next payment after the call to update the profile. Say, for example, the buyer has made one trial payment out of a total of three. `UpdateRecurringPaymentsProfile` is called to increase the number of billing cycles to five. As a result, the buyer will have four additional trial payments to make. If the call to `UpdateRecurringPaymentsProfile` is made during the regular payment period, the changes take effect with the buyer's next scheduled regular payment.

For complete details, see the *Name-Value Pair Developer Guide and Reference* or the *SOAP API Reference*.

Updating Addresses

When you update the subscriber shipping address, you must enter all of address fields, not just those that are changing:

For example, if you want to update the subscriber's street address, you must specify all of the address fields listed in the *Name-Value Pair Developer Guide and Reference* or *SOAP API Reference*, not just the field for the street address.

Updating the Billing Amount

For profiles created using Express Checkout, the total amount of a recurring payment can only be increased 20% in a fixed 180-day interval after the profile is created. The 20% maximum is based on the total amount of the profile at the beginning of the 180-day interval, including any shipping or tax amount.

For example, if a profile is created on March 10 with a total amount of $100, then during the 180-day interval from March 10 to September 6, you can increase the payment amount to a maximum of $120 (120% of $100).

Suppose that during the first 180-day interval, you increased the payment amount to $110. Then during the next 180-day interval (starting on September 7 in this example), you can only increase the amount of the payment to a maximum of $132 (120% of $110).

Billing the Outstanding Amount of a Profile

Use the `BillOutstandingAmount` API to immediately bill the buyer for the current past due or outstanding amount for a recurring payments profile.

NOTE: You can also bill the buyer for the current past due or outstanding amount for a recurring payments profile from the PayPal website.

To bill the outstanding amount:

- The profile status must be active or suspended.

  NOTE: The `BillOutstandingAmount` API does not reactivate a suspended profile. You need to call `ManageRecurringProfileStatus` to do this.

- The profile must have a non-zero outstanding balance.
- The amount of the payment cannot exceed the outstanding amount for the profile.
- The `BillOutstandingAmount` call cannot be within 24 hours of a regularly scheduled payment for this profile.

NOTE: If another outstanding balance payment is already queued, an API error is returned.

You will be informed by IPN about the success or failure of the outstanding payment. For profiles created using Express Checkout, the buyer will receive an email notification of the payment.

Recurring Payments Notifications

Merchants are notified of recurring payments events through IPN and email.

Merchants are notified of certain events through IPN. For recurring payments profiles created using Express Checkout, buyers are also notified of specific events by email. The following table indicates when IPN and emails are generated.

Recurring payments instant payment notifications and email

Event	IPN	Buyer Email
Profile successfully created	Yes	Yes
Profile creation failed	Yes	Yes
Profile canceled from paypal.com interface	Yes	Yes
Profile status changed using API	No	Yes
Profile update using API	No	Yes
Initial payment either succeeded or failed	Yes	Yes
Payment either succeeded or failed (during either trial period or regular payment period)	Yes	Yes
Outstanding payment either succeeded or failed	Yes	Yes
Maximum number of failed payments reached	Yes	No

NOTE: API transactions such as `ManangeRecurringPaymentsProfileStatus` do not trigger IPN notification because the success or failure of the call is immediately provided by the API response.

11 Using Other PayPal API Operations

You can use additional PayPal API operations to implement additional features for your buyers or to provide “back office” support.

- Issuing Refunds
- Handling Payment Review

Issuing Refunds

You can use the `RefundTransaction` PayPal API operation to issue refunds.

Use the `RefundTransaction` API to issue one or more refunds associated with a transaction, such as a transaction created by a capture of a payment. The transaction is identified by a transaction ID that PayPal assigns when the payment is captured.

NOTE: You cannot make a refund if the transaction occured more than 60 days ago.

You can refund amounts up to the total amount of the original transaction. If you specify a a full refund, the entire amount is refunded. If you specify a partial refund, you must specify the amount to refund, the currency, and a description of the refund, which is called a *memo*.

When you call the `RefundTransaction` API, PayPal responds with another transaction ID, which is associated with the refund (not the original transaction), and additional information about the refund. This information identifies

- the gross amount of the refund, which is returned to the payer
- the amount of the refund associated with the original transaction fee, which is returned to you
- the net amount of the refund, which is deducted from your balance

To issue a refund

1. In the `RefundTransaction` request, specify the transaction ID of the transaction whose payment you want to refund .

   `TRANSACTIONID=`*`transaction_id`*

2. Specify the kind of refund, which is either `Full` or `Partial`.

   `REFUNDTYPE=Full`

   or

```
REFUNDTYPE=Partial
```

3. For a partial refund, specify the refund amount, including the currency.

```
AMT=amount
CURRENCYCODE=currencyID
```

4. For a partial refund, specify the memo description.

```
NOTE=description
```

5. Execute the `RefundTransaction` operation.
6. Check the acknowledgement status in the `RefundTransaction` response to ensure that the operation was successful.

Handling Payment Review

Payment Review is a feature that identifies high-risk transactions and notifies you so that you can hold shipments until the risk has been evaluated by PayPal. You are immediately notified that a payment is under review and you should not ship merchandise or, in the case of electronic media, you should not allow download access while the payment is under review. You are notified of the resolution within 24 hours.

You can determine the status of a payment in the following ways:

- by logging into https://www.paypal.com/ and viewing the status information in the Transaction History
- by email sent by PayPal
- by Instant Payment Notification (IPN) message
- by Payment Data Transfer (PDT) message
- by checking the status of a transaction programatically

Programatically, the merchant can determine the status of a payment by checking the initial status of a transaction using any of the following the API operations:

- `DoExpressCheckoutPayment`
- `DoReferenceTransaction`
- `DoAuthorization`
- `DoReauthorization`

You can check the subsequent status of a transaction programatically by calling the `GetTransactionDetails` API operation.

NOTE: You must use version 58.0 to obtain the initial status information provided by `DoExpressCheckoutPayment`, `DoReferenceTransaction`, `DoAuthorization`, or `DoReauthorization`.

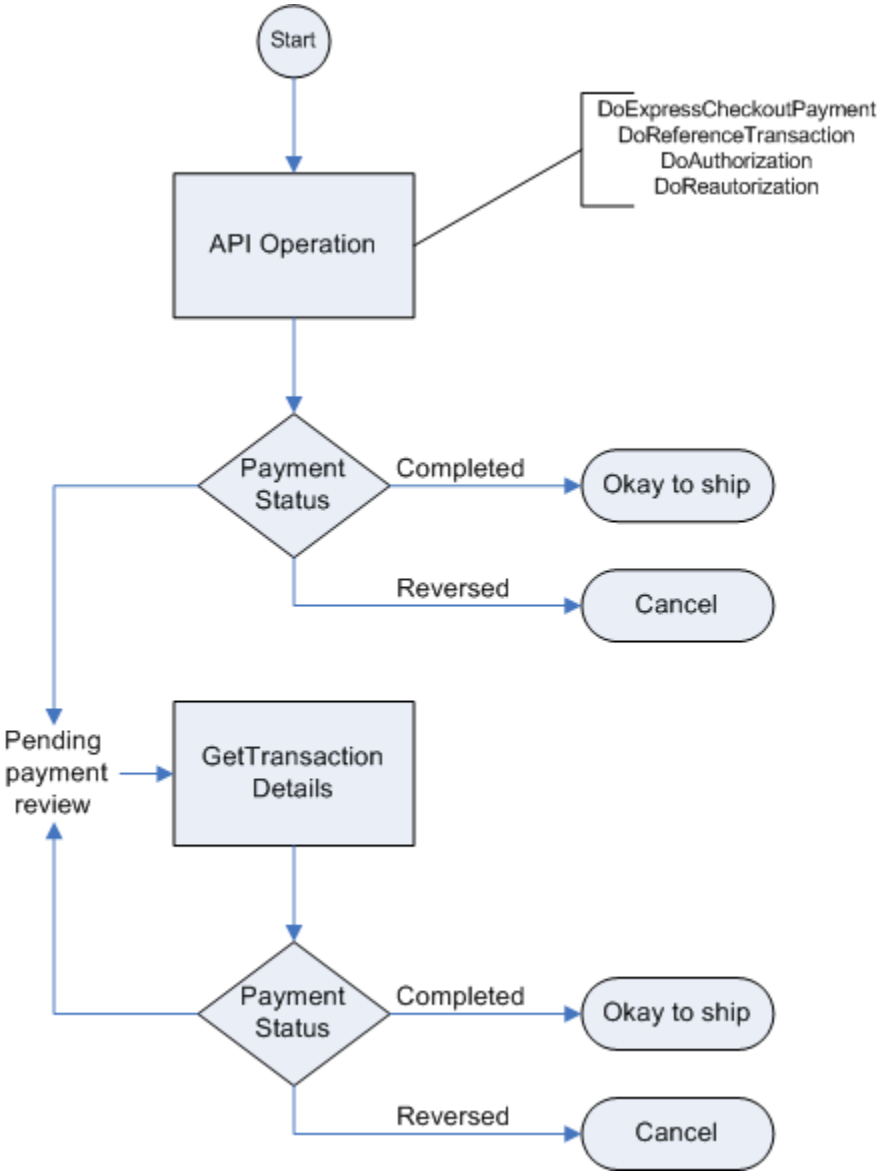
To use payment review with the `DoExpressCheckoutPayment`, `DoReferenceTransaction`, `DoAuthorization`, and `DoReauthorization` PayPal API operations, you must

1. Check the payment status in the response to the API operation; specifically, check whether `PaymentStatus` is set to `Pending`.
2. If the `PaymentStatus` is set to `Pending`, check whether the `PendingReason` is set to `PaymentReview`, because there are other reasons that a transaction may become pending. For example, an unsettled authorization's `PaymentStatus` is set to `Pending`; however, its `PendingReason` is set to `authorization`, which is not related to payment review.

If `PaymentStatus` is set to `Pending` and the `PendingReason` is set to `PaymentReview`, you should not ship merchandise or, in the case of electronic media, you should not allow download access.

Because the payment status will change after review, you must periodically check the payment status using the `GetTransactionDetails` API operation.

The following diagram shows how to use the payment status to detect payments under review by PayPal.



IMPORTANT: For best results, call the `GetTransactionDetails` API operation every six hours; never call it more frequently than once per hour.

12 Integrating giropay with Express Checkout

You must modify your Express Checkout implementation to use giropay, a common German funding source.

- giropay Page Flows
- giropay Integration

giropay Page Flows

If you accept giropay, you must redirect to the giropay website to collect the funds after completing the Express Checkout transaction.

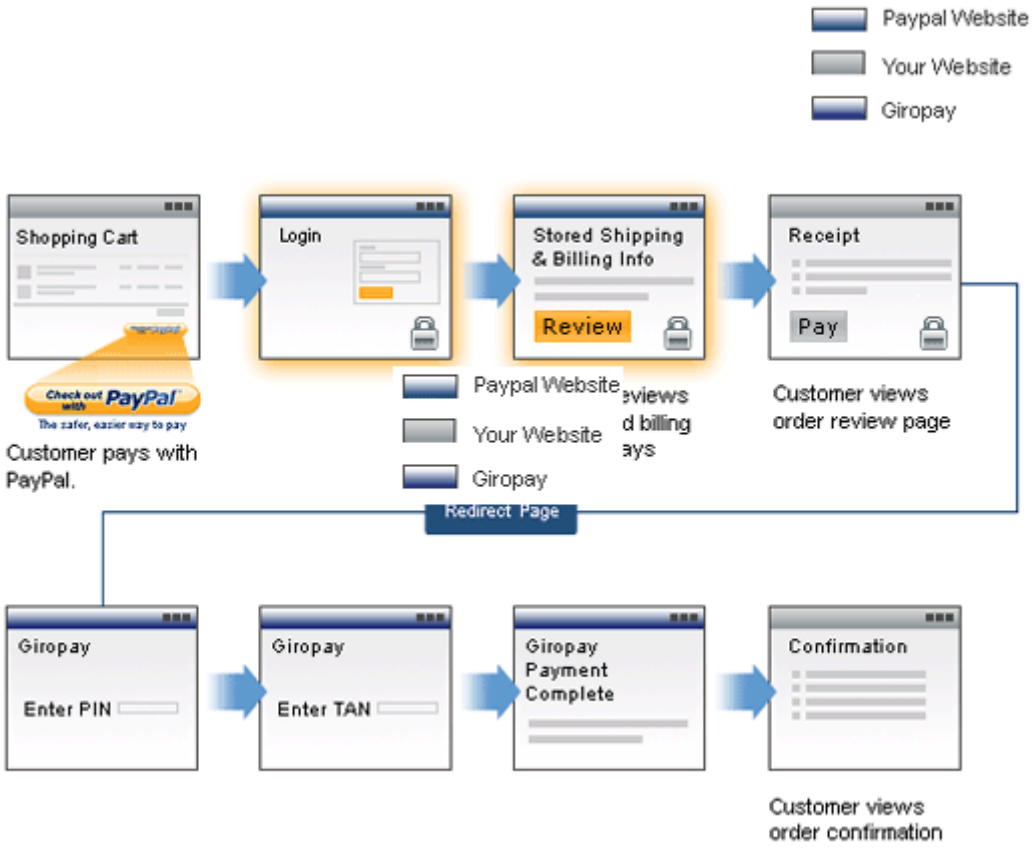
When your customer selects giropay as a funding source during the Express Checkout flow, you redirect the customer to a static PayPal URL after your order review page. PayPal then redirects the customer to the giropay website to push the funds to the merchant. After the giropay payment is successfully completed, the transaction is confirmed.

If the giropay payment fails or is cancelled by the customer, PayPal provides the necessary details for an EFT so that the customer can complete the transaction by pushing funds from his/her bank account. If your PayPal account profile blocks non-instant payments, the transactions is cancelled.

giropay Payment Page Flow

The following diagram illustrates the flow of a successful giropay payment:

Page flow for a successful giropay payment



Cancelled or Unsuccessful giropay Payment Page Flow

If the giropay payment fails for any reason, such as insufficient funds or the customer cancels, PayPal provides details to the customer to do a bank transfer from their bank account. This transaction will remain pending until PayPal receives the funds, at which time the transaction will be complete.

If you have disabled non-instant funding transactions for your PayPal account, the transaction is cancelled and PayPal redirects the customer to your *Order Cancel* page.

After the bank transfer flow is completed, the transaction is pending until the customer pushes the funds to PayPal.

If the customer cancels during the PayPal payment via bank transfer flow, your *Order Cancel* page is displayed.

giropay Integration

If you offer the giropay payment option, you must take additional steps to integrate with the Express Checkout flow.

Initiate the Flow with SetExpressCheckout

To support giropay payments, you pass the following three URLs as part of the `SetExpressCheckout` request. These URLs tell PayPal where to redirect the customer based on the success or failure of each type of payment transaction. See the *PayPal Name-Value Pair Developer Guide and Reference* for more information.

SetExpressCheckout request fields for giropay

NVP Field	Description
`GIROPAYSUCCESSURL`	The URL on the merchant site to redirect to after a successful giropay payment.
`GIROPAYCANCELURL`	The URL on the merchant site to redirect to after a giropay or bank transfer payment is cancelled or fails.
`BANKTXNPENDINGURL`	The URL on the merchant site to transfer to after a bank transfer payment.

Redirect the Customer to PayPal

After selecting a funding source on PayPal, the customer is redirected back to your website, as in the regular Express Checkout flow. There is one additional field, `REDIRECTREQUIRED`, returned in the response from both `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

If the value of this field is `true`, you redirect the customer from your *Order Review* page to `https://www.paypal.com/webscr?cmd=_complete-express-checkout&token=<token>`. PayPal then redirects the customer from this redirect page to the necessary page for the selected funding source.

GetExpressCheckoutDetails and DoExpressCheckoutPayment response field

NVP Field	Description
`REDIRECTREQUIRED`	Flag to indicate whether you need to redirect the customer to back to PayPal

The `GetExpressCheckoutDetails` response contains the `REDIRECTREQUIRED` field, which lets you know if you need to redirect the user after your *Order Review* page. You can use this value to inform the customer on your *Order Review* page that they will be sent to the giropay website to complete the order.

Complete the Transaction

Corresponding to the three fields passed to `SetExpressCheckout`, you must add the following three additional pages to your website:

Additional pages required for giropay integration

NVP Field	Description
Order Completion	The `GIROPAYSUCCESSURL` page that PayPal redirects the customer to after a successful giropay payment.
Order Cancellation	The `GIROPAYCANCELURL` page that PayPal redirects the customer to after a giropay or bank transfer payment is cancelled or fails.
Order Pending	The `BANKTXNPENDINGURL` page that PayPal redirects the customer to after a bank transfer payment.

Receive Transaction Status Notification

After PayPal redirects the customer to giropay, you receive transaction status information in the following ways:

- IPN Notification
- Email (only for successful giropay or bank transfer transactions)
- PayPal Account Overview
- PayPal reports